A Companion Robot for Modeling the Expressive Behavior of Persons with Parkinson's Disease

A dissertation

submitted by

Andrew P. Valenti, BS, MS, New York University

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

May 2020

ADVISOR: Matthias Scheutz

The Dissertation Committee for Andrew P. Valenti

certifies that this is the approved version of the following dissertation:

A Companion Robot for Modeling the Expressive Behavior of Persons with Parkinson's Disease

Committee:

Matthias Scheutz, Supervisor

J.P. de Ruiter

Gina Kuperberg

Antonio Roque

Matthew Marge

To Dr. Robert B. K. Dewar who initially encouraged this fledgling undergraduate student to consider a graduate career in Computer Science and whose effortless and intuitive programming and teaching style continues to motivate me to this day. Most importantly, this work is also dedicated to my wife, Gail, without whom I would not have had the courage to return to graduate school and the drive to complete this journey.

Preface

All work presented in this dissertation was conducted at the Human-Robot Interaction Laboratory at Tufts University under the supervision of Dr. Matthias Scheutz.

This dissertation includes work that was previously presented in conference papers or journal articles, or will be presented in conference papers and journal articles.

- **Chapter 3** contains text and/or content from a paper published in the journal *Assistive Technology* [Valenti et al., 2019c].

- **Chapter 4** contains text and/or content from a paper published in the proceedings of the *11th International Conference on Social Robotics* [Valenti et al., 2019a].

- **Chapter 5** contains text and/or content from a paper to be published in the proceedings of the *The 13th PErvasive Technologies Related to Assistive Environments Conference (PETRA '20)* [Valenti et al., 2020a].

- **Chapter 6** contains text and/or content from a paper submitted to the *The 21st Annual SIGdial Meeting on Discourse and Dialogue* [Valenti et al., 2020b].

- **Chapter 7** contains text and/or content from a paper published in the *Proceedings of the 12th International Conference on Cognitive Modeling* [Valenti and Scheutz, 2013].

- **Chapter 8** contains text and/or content from a paper published in the *Proceedings of the 38th Annual Conference of the Cognitive Science Society* [Valenti et al., 2016], and a paper published in the *Proceedings of the 16th International Conference on Cognitive Modeling* [Valenti et al., 2018].

Acknowledgments

I was fortunate to consider my return to graduate school to complete my PhD at the time that Tufts created the Doctoral program in Cognitive Science. The program provided me with the opportunity to compare and contrast the mathematical and computer science notion of intelligence with the psychological and philosophical theories of human cognition. This dissertation emerged from the rich, interdisciplinary brew of insightful discussions with the research staff at Human-Robot Interaction Laboratory, mixed with collaboration with the research staff in the Psychology and Occupational Therapy departments at Tufts.

First, I would like to thank my primary advisor Matthias Scheutz for keeping me on track throughout my academic career at Tufts, for his encouragement during the inevitable frustrating moments, and for the opportunity to work on the several interesting research investigations which allowed me carve my own direction and develop the intuition required to formulate and seek the answers to research questions. Without his help and support, I would have not been the researcher, writer and instructor that I am today. I also would like to thank my secondary advisor Gina Kuperberg for her seemingly limitless belief in my abilities and for the good fortune that I was able to take her courses which stimulated my interest in The Predictive Mind theory and influenced my research direction. In addition, I had the good fortune to work with Laura de Ruiter whose knowledge of the psychology of bilingualism was a great help, and Linda Tickle-Degnen, whose work with persons living with Parkinson's disease contributed an essential foundation to the model.

I would also like to thank the other members of my dissertation committee. J.P. de Ruiter was a mentor and collaborator during my teaching assistant assignments and who always challenged me with thought-provoking questions and suggestions. Matt Marge and Antonio Roque carefully reviewed my dissertation and provided feedback on subtleties of my research and found all those typos which greatly strengthened and polished the final version.

At the Human-Robot Interaction Lab, I was fortunate to cross paths with

many outstanding graduate students whose work set an example for me and helped remove the clouds obscuring the solutions to some particularly stubborn problems. More specifically, I would like to thank a number of them with whom I had the most interaction including, Meia Chita-Tegmark, Felix Gervits, Daniel Kasenberg, Brad Oosterveld, Sepideh Sadeghi, Vasanth Sarathy, and Ravenna Thielstrom. I would like to thank all my interns whose tireless coding efforts gave me the space and time to work on the hard researcher problems: Avi Block, Alex Bock, Derek Egolf, Michael Gold, and Morgan Ciliv.

Finally, I would like to thank my wife Gail, my daughter Alessandra, and my son Drew. My family always believed and supported me even when I wondered whether returning to graduate school was the "right" decision. Without their encouragement, patience, and sacrifice it would not have been possible for me to complete this dissertation.

A Companion Robot for Modeling the Expressive Behavior of Persons with Parkinson's Disease

Andrew P. Valenti

ADVIS0R: Matthias Scheutz

Emotions are crucial for human social interactions and as such people communicate emotions through a variety of modalities: kinesthetic (through facial expressions, body posture and gestures), auditory (through the acoustic features of speech) and semantic (through the content of what they say). Sometimes however, communication channels for certain modalities can be unavailable (for example in the case of texting), and sometimes they can be compromised, for example due to a disorder such as Parkinson's disease (PD) that may affect facial, gestural and speech expressions of emotions. As a result, it is not easy for caregivers to judge how PD persons are coping with their condition. They may look as if they are unfeeling, indifferent, sad or hostile and misinterpretation of their true internal state can lead to depression.

In this dissertation, we present a situated emotion expression framework which a robot can use to detect emotions in one modality, specifically in speech, and then express them in another modality, through gestures or facial expressions. This is part of a larger objective to develop a socially assistive robot for the social self-management of people with PD. The framework compensates for ambiguities in natural language, disfluencies that are often present in the speech of persons with PD, and errors in the automatic speech recognition system. More generally, the framework would be useful for any conversational AI agent and we demonstrate ways in which it can be extended to a bilingual environment. Finally, we demonstrate a model of human language processing that can be used to monitor human-level performance using a biologically-plausible model that uses dynamic neural fields.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Table 1.1: Dissertation scope.

	Overall project	This dissertation
Objective	Develop a socially assistive robot for the social self-management of health of people with PD.	Develop a prototype for an emoting robot that can detect emotions in one modality (spoken language) and express them in another (gestures).
Robot capabilities	Simple interaction, observation of activities, mediation of interactions.	Detection and expression of emotion.
Hypothesis	The assistive robot will reduce stigma and improve communication between people with PD and caregivers and healthcare providers.	Robust emotion expressions of the robot can be correctly perceived in both high and low frequency emoting conditions.
Validation	Clinical trial with people with PD and their caregivers.	General population studies with robot emoting in different conditions.

Persons living with Parkinson's disease (PD) often exhibit facial masking (hypomimia), giving them a fixed, mask-like expression. This symptom results in a disassociation between their true mental state and outward expression making it difficult for those who interact with them to infer their true mental state. Ascertaining their true state can help guide interaction which can improve the person's

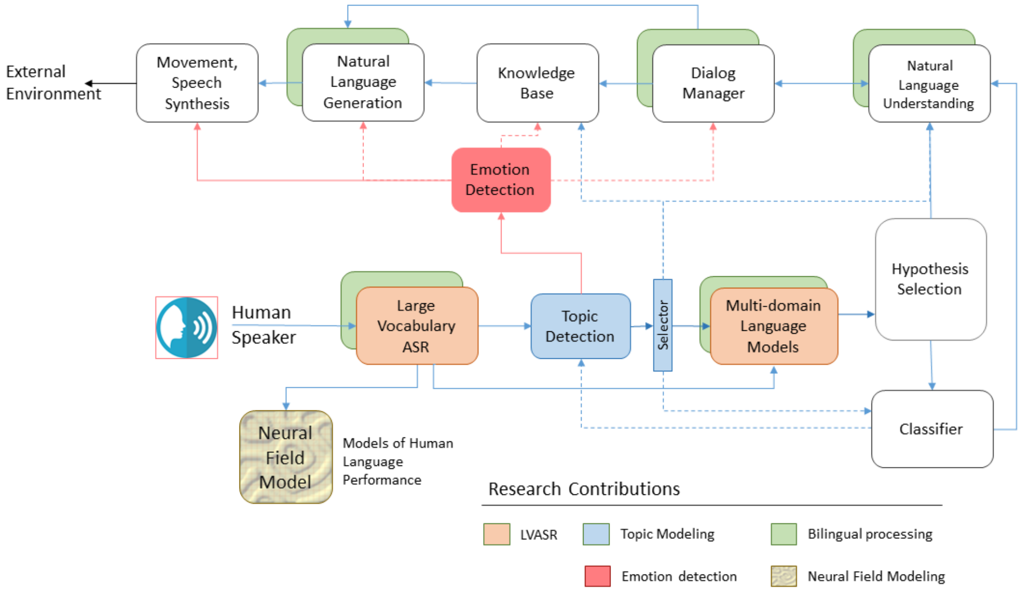


Figure 1.1: The Situated Emotion Expression Framework consists of the Large Vocabulary Automatic Speech Recognizer (LVASR) which uses Topic Detection to automatically select from among the Multi-domain Language Models. Transcribed speech is passed to the Emotion Detection component which forwards its prediction to the components as shown. Bilingual processing is implemented in the indicated green components. Models of language performance can be implemented using a Neural Field Model, which receives input from the LVASR. Dotted-line connections indicate proposed functionality that has not yet been implemented.

motivation in social and therapeutic situations. We suggest that an emoting robot can help people with PD express their inner emotional states and thus improve their communication with caregivers. In this dissertation, which is part of a larger project (see Table 1.1), we develop an *Situated Emotion Expression Framework* for a socially assistive robot which is robust to errors and can automatically infer the mental state of a PD person from their speech (see Figure 1.1). We also demonstrate how this framework can be extended to measure human language performance and operate in a bilingual environment.

The objective of this dissertation is a step towards the longer-term goal in which the framework can be embedded in an intelligent agent to monitor the interaction between the person and, for example, a caregiver. This agent might also be used in clinical settings to assist the occupational therapist in, for example, patient evaluation. The framework is not restricted to the domain of persons living with PD; it should be able to generalize and serve as an intelligent agent useful for detecting emotion and generating appropriate responses in the interaction between the intelligent agent and the human. Research in the area of "affective computing", has been shown this ability to increase user engagement, a desirable goal for applications including, health-care, tutorial systems, travel, financial services, and games.

This dissertation is organized in three parts. In the first part, we develop, implement, and evaluate the theories which form the basis of the situated emotional framework. In the second part, we discuss related research and demonstrate how it can be used to usefully extend the framework. In the last part, we present our conclusions.

1.1 Primary Contributions

1.1.1 Inferring Emotional State in Persons with Parkinson's Disease

In Chapter 3, we lay groundwork for detecting sentiment in the continuous speech in persons with PD by demonstrating the efficacy of using the Latent Dirichlet Allocation (LDA) generative model to extract topic proportions from a collection of written text documents which are then used as input features to train a classifier to detect positive and negative sentiment. We chose LDA as a principled way to learn features, unsupervised in the expectation that it will better generalize to other domains than other approaches which use hand-engineered features or sentiment lexicons (e.g., Pennebaker's Linguistic Inquiry and Word Count, "LIWC"). Takahashi and Tickle-Degnen have shown that LIWC's measure of positive and negative emotional valence correlates to the words a PD person uses when asked to recount an enjoyable and frustrating experience. Since the goal of our research was to measure emotional content as best we could at the utterance level, as speech unfolds, we evaluated the performance of LIWC and LDA for short sentences, i.e., average word count of 13 (typical of spoken utterances) and found LDA to outperform LIWC.

1.1.2 Detecting Emotion Polarity from Continuous Speech in a Robot

In Chapter 4, we use the experience gained using LDA to detect sentiment on the document level to develop a version that can predict the emotion on a sentence by sentence basis. In order to train the model, we need to collect written text which is labeled incrementally (e.g., sentence by sentence). To this end, we built a Web-based tool called the Emotional Inference Topic Model (EMIT) which presents lines of text one at a time to a human evaluator. We next integrated the emotion detection model in the DIARC robotic cognitive architecture which then gave a robot the ability to express emotion using facial expressions, i.e., Robo-Motio's Reddy robot. We defined a three-state emotion detector that can predict positive, negative, and neutral emotion from utterances that have been transcribed from the PD interviews and used it to drive three facial expressions on the Reddy robot: smile, frown, and neutral. We tested this using the ground truth labels obtained from the human evaluators.

We extended this work by incorporating a Large Vocabulary Automatic Speech Recognizer (LVASR) into DIARC in such a way that it can transcribe ordinary conversational speech in real time and present the transcribed utterances to the emotion detector. We used an already-trained acoustic and language model that was developed using the Kaldi toolkit. We evaluated the entire emotion pipeline by using a human speaking to the robot and conducting a human-robot interaction experiment to evaluate how well the robot's facial expressions match was it is being said.

1.1.3 A Dynamical System for Expressing Fine-Grained Emotions in a Robot

In Chapter 5, we note that a three-state emotion model may not be sufficient to capture and express some of the more subtle human emotions. In the fourth part, we expanded the three-state model to five-states. We further used the more fluid bodily movements available through gestures in SoftBank Robotic's Nao robot. We developed a means to express increasing levels of emotional positivity using seven gestures (five detected emotions plus two transitional gestures). We conducted an HRI experiment to evaluate these seven new gestures to ensure that an observe can recognize that they express increasing levels of positivity. In this part of our research, we also identify and address the challenge of handling LVASR and emotion detector performance. We build an expressor component that takes as input predictions from the detector and uses the dynamical properties of a mass-spring model for smooth transitions between emotion expressions over time. This novel method compensates for varying utterance frequency from the LVASR and prediction errors coming from the emotion recognition component. We conducted an HRI study to validate how well the robot's gestures matched what is being said in the with- and without- spring model conditions by varying specific emotion detection errors.

1.1.4 Improving Natural Language Understanding in Spoken Dialog Systems

In Chapter 6, we demonstrate a novel framework for understanding spoken dialog in which utterance analysis is escalated through a multi-level system *according to the feedback retrieved at the syntactic, semantic, and contextual/topic level.* Analysis is applied incrementally at each level as the system attempts to resolve the uncertainty surrounding utterance interpretation. We evaluate the accuracy of its semantic interpretation of user utterances in two task domains against a control without such a mechanism. We demonstrate how this system can extend the situated emotion expression framework.

1.2 Related Research

1.2.1 Building a Bilingual Robot

In Chapter 7, we investigate and demonstrate two computational models that further our understanding of how to extend the model framework to a bilingual environment. The first is a computational model of the inhibitory control theory which states that the non-target language of the bilingual is suppressed by top-down contextual cue. The second is a computational model of bilingual memory which describes a psychological theory of how words from the multiple languages interfere with each other and how the desired word is selected using the top-down control mechanism. Similarly, we demonstrate which parts of the model framework need to be modified so that it can freely switch among syntactic and semantic components according to the task demand and correctly interpret the meaning of the human utterance in the target language.

1.2.2 Modeling Human Language Processing with Neural Fields

In Chapter 8, we introduce a biologically plausible model of human language processing, the dynamic neural field and discuss two investigations. In the first, we correlated the dynamics of the field to one measure of human language performance, word repetition under two workload conditions. In the second, we demonstrated how neural fields can be connected to create a computational model of the Cohort theory of word recognition. We demonstrate how the neural field can be used in the framework to measure human cognitive performance from the speech signal and how this information can be used to monitor the workload and adjust task performance.

Chapter 2

Sentiment Analysis

The design of the emotion detection component of the framework was informed by prior investigations in the field of sentiment analysis. We designed the framework for a particular application, a companion robot for persons living with Parkinson's disease. More generally, it can be applied to conversational agents which, like the robot, assumes human interact with it using conversational speech. However, interaction may also be through conversational text. In this chapter, we survey sentiment analysis as applied to conversational agents.

This chapter makes the following contributions. We systematically organize and review the existing literature on sentiment analysis models and develop a framework for analyzing them. We describe the psychological models of emotions We explain the most common approaches and challenges they were intended to solve. Finally, we indicate which models are freely-available to the researcher or are available for a fee.

2.1 Introduction

Conversational agents may be defined as software systems which interpret and respond to statements made by users in ordinary natural language [Ram et al., 2018]. These agents, messaging apps, speech-based assistants ("smart speakers"), socialbots, and chatbots, are becoming increasingly popular in health-care, travel, financial services, gaming, and other market segments where they engage the user through voice or chat in order to understand and correctly act on their inputs in real time. As the technology for wearable devices continues to evolve, it is likely they too will have some form of conversational agents. Conversational agents are computational components that may receive either spoken or text input and generate the sentiment predictions incrementally, in real-time, without relying on facial affect and gestures. One way to generate engaging responses is by taking into account and responding to the sentiments explicitly or implicitly expressed by users. This can increase user engagement [Fraser et al., 2018] or help the system to better understand the user's intent and generate an appropriate response.

Rosalind Picard, founder and director of the Affective Computing Research Group at the MIT Media Lab goes a step further and argues that in human cognition, thinking and feeling are both present and there is a reciprocal relationship between the human's neurological center of emotion, the limbic system, and the center of thinking, the cortex [Picard, 1995]. Thus, emotions play an essential role in rational decision-making, perception, and problem solving. The implication is that expressing and recognizing affect is important for human-computer interaction because it allows conversational agents to be able to respond to the evolving human affective state by modulating their responses and expressing their emotions. One example where this can be useful is a tutorial system which dynamically adapts its lessons to the students' affective state, providing an easier approach if it detects that the student is frustrated.

However, there are differences when detecting sentiment in conversational speech and written text. Unlike detecting sentiment in certain social media (e.g., tweets, movie and product reviews, news opinion articles), unconstrained natural language may contain no discernible ground-truth, making it difficult to train such systems and evaluate how well they perform during a human conversation. In addition, the speech recognition system may generate errors or segment the speech into small chunks, both outcomes presenting a challenge to sentiment detection. In Section 2.3, we review some additional characteristics of human conversation that might constrain the performance of the sentiment detection model.

In this chapter, we review the models, methods and limitations of the most common approaches for sentiment analysis. We characterize the articles according to how they assume sentiment is represented, i.e., as a bag-of-words, sentiment lexicon, topics, semantics, syntactic structure, or pre-trained language models. We describe the algorithms used to extract sentiment from these. We describe the dataset, experimental procedures and the performance metrics used to validate the models and their limitations. We indicate which papers describe models that have a reference implementation or make the source code freely available so that it can incorporated in the researcher's own work. This review does not attempt to be an exhaustive survey of sentiment analysis as progress in this area is fast-paced. It aims to illustrate the most widely-cited approaches for detecting sentiment analysis that can be used by conversational agents.

Detecting sentiment in ordinary natural language comes down to deciding which input modalities can be analyzed reliably. Assuming that the facial modality is not available in conversational agents (thus excluding devices such as the Facebook Portal, Amazon Alexa Show, and Google NestHub Max), the audio signal from the voice channel is available as input. One challenge in using using the speech signal directly is that different users have different physiological responses to the same emotional state; for example, they are not reliable in individuals with certain clinical conditions such as Parkinson's disease. Another are privacy concerns in which speaker identification may be performed using the acoustic characteristics contained in the voice signal.

This chapter assumes the existence of an Automatic Speech Recognizer (ASR) that converts, as accurately as possible, the speech signal to text. Alternatively, the user may communicate with the system via a text input system such as a keyboard. As a result, this review covers detecting sentiment in text, whether in transcriptions of natural human speech or typed; emotion modeling is much wider and includes using audio speech, gestures, and facial expression modalities either singularly or in combination to detect emotions.

This chapter proceeds as follows. We give some basic definitions of commonly used terms in the emotional modeling literature, indicate which are used ambiguously, and describe terms we will use consistently in our review. The sentiment analysis computer-based systems (i.e., "computational models") we review were based on one or more psychological models of emotion. Therefore, we introduce the three major traditions of psychological emotion and give some examples of computational models that were influenced by a particular tradition. We then briefly review the differences between detection emotion in typewritten text and transcribed speech.

We then review the computational modeling approach to emotion detection. We believe that an excellent way to understand the various models is to categorize them according to the assumptions they make regarding the how emotion is represented in the language, e.g., by a bag-of-words, sentiment lexicons, latent topic structure, semantic properties, syntactic structure, or through deep learning of the features themselves. As a result, the reviews are organized according to these features. Although we are primarily interested in detecting emotion in conversational text, we briefly review related work which uses spectral features of the speech signal. We do so to gain a perspective as to how critical this information might be to emotion detection. We conclude by reviewing the main challenges for sentiment analysis and summarizing what we have learned.

2.1.1 Background

Traditionally, human communication may be partitioned to verbal and nonverbal channels. The nonverbal channel may be subdivided into the paralinguistic (i.e., speech characteristics) and visible (i.e., facial expression, gestures) channels. It might be assumed that the two nonverbal channels are important in communication of affect and a study by Mehrabian [1972], the researchers estimated that about 7% of emotion is communicated via the verbal channel. However, this did not use situations normally arising in natural conversation and the researchers cautioned against generalizing their findings. In fact, in another set of experiments conducted by Krauss et al. [1981], the researchers found no support for the assumption that nonverbal communication was the primary basis for communication of affect. In one of the experiments, the researchers found that verbal information was the largest single factor in evaluative judgments of affect. Even so, they caution that judging affect may depend on a number of factors. For example, nonverbal channels may take on more importance when gross discrepancies exist among channels; however this may not be the case otherwise. In general, though, Krauss et al. [1981] conclude that the common assumption that nonverbal communication is the primary source of the human ability to infer affect does not appear to be true.

Pang et al. [2008] provide an excellent discussion of how the field of sentiment analysis evolved. This paper has been cited over 9,000 times and has influenced most, if not all, of the models we review. The authors approach sentiment analysis by analyzing responses to the question, "What do others think?" According to the authors, prior to the pervasive reliance on the Internet as a means for obtaining information that might help us, for example, buy a car, decide whom to vote for, where to eat, we most likely relied on friends, colleagues or product review publications, e.g., *Consumer Reports.* At present, more and more people are willing to rely on non-professionals for these decisions through the use of crowd-sourced, online rating systems, which in many cases is in the form of unstructured text. Many times though, the information is incomplete, confusing, hard to find, or overwhelming. As a result, a research motivation is to create better systems that can automatically sift through unstructured information and extract opinions on various aspects of a product or service.

Opinion mining of text can be a challenging problem in natural language processing, and is often used to push the state-of-the-art forward. For example, identifying which aspect of a product is favorable or unfavorable is goal of the research competition such as SemEval-2014 Task 4 in which the tasks is: given a customer review, determine the aspect terms, aspect categories, and sentiment towards these aspect terms and categories [Manandhar et al., 2014].

On the other hand, for a conversational agent which may expect input on any topic, a sentiment system that generalizes over many different domains is more useful than a a model which extracts opinions from all aspects of a movie review, for example. However, Pang and Lee use opinion mining as the basis for discussing the challenges of extracting and processing opinions from subjective information and many of these are the challenges driving research and development of most recent models; thus we include these in our review. Moreover, we found that their discussion of how salient information (i.e., emotion) in text can be represented in several different ways as features, is useful in categorizing different sentiment analysis models. We found this to be intuitive way to categorize the types of models discussed in this review.

2.1.2 Basic Terminology

In this section we discuss research papers which sometimes uses terms interchangeably even though they have separate meanings. In this section, we define commonly used terms and strive to offer a single consistent definition, according to where the majority of our surveyed papers agree. We point out terms for which there seems to be multiple usages.

Two frequently used terms in the literature are *sentiment analysis* and *opinion mining*. According to the definition given in the Merriam-Webster Online Dictionary [bya, 2020], opinion and sentiment are synonyms and mean a judgment one holds as true. However, this does not imply they have identical meanings. Opinion is used when the judgment is not yet final or certain but is founded on some facts. Sentiment suggests a settled opinion reflective of one's feelings. Opinion mining is a popular term within the Web search and information retrieval technical communities since its appearance in a paper by Dave et al. [2003].

Sentiment analysis arose in the Natural Language Processing (NLP) communities where it was used to reference automatic analysis of text using natural language processing in papers such as Das and Chen [2001] and Nasukawa and Yi [2003]. In this context, sentiment analysis most often refers to the techniques used to infer the binary emotional polarity (e.g., positive, negative) of a person as they

Table 2.1: Basic terminology

	Definition	Resources
Affect	Frequently used in the psychological and computational literature to ground the more “fuzzy” concept of emotion when describing a model of emotion. In other definitions, affect subsumes concepts not traditionally considered emotion such as mood.	[Russell, 1980]
Affective Computing	Computing that relates to, arises from, or influences emotions.	[Picard, 2000]
Arousal	A physiological measure of how calm or excited a person is, but can also be assessed subjectively via self-report.	[Reisenzein, 1994, Picard, 2000]
Computational Model	A technical design and/or computer-based implementation that embodies descriptions of psychological cognitive processes.	[Sun, 2008]
Emotion	Originally: an agitation of mind; an excited mental state. Subsequently: any strong mental or instinctive feeling, as pleasure, grief, hope, fear, etc., deriving esp. from one’s circumstances, mood, or relationship with others.	[OED, 2020, Smith et al., 1990]
Sentic computing	An approach which relies on the ensemble application of common-sense computing and the psychology of emotions to infer affect in natural language.	[Cambria and Hussain, 2012]
Sentiment analysis	Typically refers to the detection of the binary emotional polarity (positive or negative) of text, but may also include additional categories.	[Pang et al., 2008]
Valence	A subjective measure of positivity, often on a scale from displeasure to pleasure, intended to evaluate the individual’s response to emotion-eliciting circumstances, or to measure subjective feelings or attitudes.	[Harmon-Jones et al., 2011]

interact with a text or a document, rather than determining the specific human emotion Pang et al. [2002]. However, sentiment analysis approaches may also attempt to classify additional emotion labels. Rather than recognizing emotions as belonging to discrete and often binary categories, *emotion recognition* attempts to infer a set of emotion labels such as happiness or satisfaction, both of which fall under the positivity category. When categorizing the approaches in the papers we reviewed, we follow Pang and Lee's broad interpretation, and use the terms "sentiment analysis", "opinion mining", and "emotion recognition" interchangeably [Pang et al., 2008].

Sentiment analysis is also connected to *affective computing*. Affective computing is a term that Picard [1995] used to describe "computing that relates to, arises from, or influences emotions." It is a cross-disciplinary field that covers not only sentiment analysis, but emotion detection, interpretation, and simulation. In general, it is concerned with how affective factors condition interaction between humans and technology and, conversely how affective sensing and simulation techniques can inform the understanding of human affective processes. Thus, many of the papers we reviewed situate themselves in the broader area of affective computing.

In a later section we will introduce the three traditions of psychological emotion theory which provide the frame work for the papers we reviewed. We shall use the term *computation models* to firmly distinguish the sentiment analysis technical design and implementation from the the underlying psychological models of emotion.

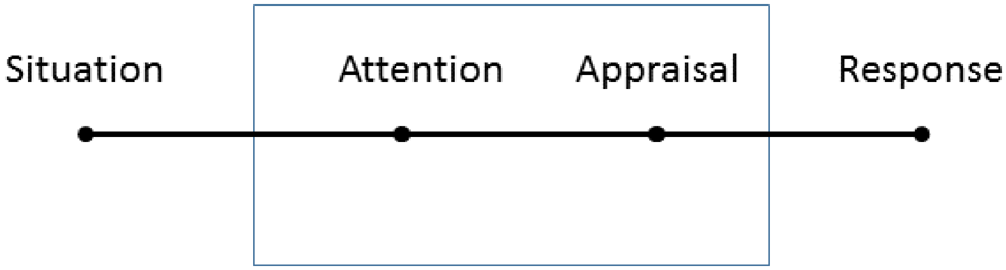


Figure 2.1: The "modal model" of emotion.
[Gross and Thompson, 2007]

2.2 Psychological Models of Emotion

Computational models of sentiment analysis must make some assumptions about the nature of emotion, e.g., are emotions to be classified categorically or are they real numbers along a dimension and, if so, how many dimensions? Since the purpose of this paper is to review models suitable for conversational agents, we would like them to understand and express emotions similarly to way the human would. Thus in this section we will briefly discuss emotion theory and the major traditions from which they arise.

In the early part of the twentieth century, psychologists defined emotion based on observed specific behavior and physiological changes of the emotion expression (for a good historical perspective, see [Smith et al., 1990]). In this definition emotion is a complex behavioral phenomenon involving many levels of neural and chemical integration [Lindsley, 1951]. However, as Fehr and Russell [1984] reported: "Everyone knows what an emotion is until asked to give a definition." While there is agreement among cognitive psychologist that states such as anger, sadness, and fear can be regarded as emotions and others, such as hunger and thirst should not, there are other states where there is little agreement, e.g., startle, interest, guilt [Ekman, 1984, Plutchik, 1984]. It is difficult to define the necessary and sufficient conditions that would constitute something as being an "emotion". Emotion theories attempt to answer this question and to explain its purpose as a human condition. As of yet, there is no single, unified theory as they have emerged from the writings of different psychologists. From a historical point of view, the major contributors to our thinking about the nature of emotions are Charles Darwin, William James, and Sigmund Freud. Charles Darwin was the first major researcher to hypothesize on the nature of emotions.

Darwin had come to recognize that the concept of evolution applied not only to the the physical but also to animal behavior. In addition to intelligence, memory, and reasoning, the emotions expressed by humans and lower animals have also evolved as a mechanism to allow the survival of most animal species [Darwin and

Prodger, 1998]. Early in the evolutionary cycle these adaptations were more rigid and reflex-based and emotions represented a move away from built-in responses to environmental stimuli to a more flexible, complex and variable behavior. During this evolutionary process, thought and judgment "emotional patterns" filled the gap between environmental stimuli and action, allowing higher species to survive by learning how to deal with their environments [Lazarus and Lazarus, 1991]. The seven basic emotions described by Ekman and Cordaro [2011], (i.e., anger, fear, surprise, sadness, disgust, contempt, happiness) are derived from the facial expressions of emotion reported by Darwin. These influenced several approaches to the computational models of sentiment analysis some of which use all or a subset of these terms as their output.

Aside from the evolutionary tradition, there are two other major directions from which to approach emotion theory. William James, a late nineteenth century American psychologist developed the *psychophysiological* context which studies the relationships between subjective feelings and physiological states of arousal [James, 1894]. In the purest form of James' theory, the bodily response to stimuli *is* the emotion, which is not accepted today. If one considers that people can experience emotion without a corresponding physiological function (e.g., love), we begin to appreciate how this may be complicated by a number of factors. For example, the intensity of the emotion, its type (e.g., love for a particular food vs. romantic love), the inducing state of the emotion, how the person expresses emotion [Picard, 1995, 2000]. We will later see the concept of arousal used in computational models of emotion detection. Lazarus and Lazarus [1991] present a cognitive approach to emotions called *appraisal theory.* In this theory, cognitive appraisals are precursors of all emotional states. Thus, as a person experiences an event their thoughts must precede the arousal and emotion, which happen simultaneously. The theory is the basis for the computational model described in Balahur et al. [2011].

The third tradition is the *dynamic context.* This tradition is identified with Sigmund Freud and suggests that emotions are part of a person's biological nature but can undergo a large variety of transformations during the course of the person's life. Theories drawn from this tradition do not underlie the computational models reviewed in this paper. For a comprehensive discussion on theories of emotion, see [Plutchik and Kellerman, 2013].

Gross and Thompson [2007] give a highly abstract yet intuitive modal model of emotion (see Figure 2.1): a person-situation transaction that compels attention, has particular meaning to an individual, and gives rise to a coordinated yet flexible multi-system response to the ongoing person-situation transaction. This model is consistent with Gross and Thompson's model does not contradict these theories and we find its transactional nature is similar to the way a conversational agent operates.

The emotion labels to be detected computationally in text may be derived from the various psychological theories of emotion. For example, Ekman [1992], argues that humans have evolved emotions as a means for adapting to their environment. In his theory, there are six basic emotions which all humans share: happiness, sadness, fear, anger, disgust, and surprise. Plutchik [1980] also argued for the primacy of emotion for evolutionary survival. In his theory there are eight primary emotions: anger, anticipation, joy, trust, fear, surprise, sadness and disgust. Some are polar opposites of one another (e.g., joy-sadness) and emotions can be combined and their intensity measured. This trend towards measurability and continuity in emotional models was solidified by Russell's Circumplex Model of Affect [Russell, 1980]. In this model, emotions are arranged in a circle around two axes, arousal and valence. This created a continuous space in which emotions could be plotted according to their value along these two scales. Thus, there are some computational models which attempt to detect continuous values of emotional valence or arousal e.g., [Hutto and Gilbert, 2014, Tausczik and Pennebaker, 2010a, Socher et al., 2013].

2.3 Detecting Emotion in Naturally Occurring Conversations

This review covers the methods and models used to extract sentiment (fine-grained and coarse) in conversational agents and, as such, conversations can originate from speech or from typing. Depending on the originating modality, the efficacy of detecting emotion is likely to be very different. For example, in business email communications, people are likely to perceive negative emotions with greater intensity than they do positive emotions [Byron, 2008]. And in social media, people present different online identities that impact the impression that others have of them [DiMicco and Millen, 2007]. Many people read a message and infer the emotions that are conveyed by the sender. The challenge is whether an conversational agent can detect the emotions that are disclosed by a message accurately and automatically. Opinion mining of product reviews and social media are practical applications of this challenge and they have motivated much of the research behind computational models of sentiment detection. On the other hand, for some conversational agents, e.g., Amazon Alexa, improving human interaction with the agent while using naturally occurring conversation is an active research project Ram et al. [2018].

However, there are three major differences between text and in-person interaction: (1) time-sensitivity (i.e., text is less sensitive than in-person), (2) interactivity (i.e., during in-person interaction, the listener can change the message as it comes out), and (3) incrementalism (i.e., text interaction largely appears all at once). From these, we can infer that for naturally occuring conversations, agents must be able to process continuous input in real time while updating their sentiment predictions incrementally. We now examine some of the challenges for detecting emotion in text originating in naturally occurring conversation, drawing from the field of Conversational Analysis (CA), a qualitative method for studying human social interaction. For this discussion, we review the works of Edwards [1999] in *Emotion Discourse* and Hoey and Kendrick [2017] in *Conversation Analysis.*

As discussed in Section 2.2, emotions are complex reactions stemming from natural bodily experiences, older than language and express genuine feelings rather than thought. Edwards [1999] writes that in interpersonal, naturally occurring conversations, emotions are invoked and descriptively built in such conversations. Emotion discourse analysis gives us a feel of how events and a person's state of mind leads them to *talking* of temporary mental states by using emotional categories such as “angry” or “jealous” to situate the conversation. Rather than placing the focus on the semantics of the emotion word a person uses, the emphasis now is on what people are doing when they use these words. In his paper, Edward describes how emotion are a means for studying how actions, reactions, motives, dispositions and other psychological categories are assembled as part of a narrative and can explain human conduct. In this view, emotional mental states do not cause what a person talks about; the categories and concerns of the discourse are a reflection of their mental state.

We draw two conclusions from Edwards’ paper. An emotional state can be inferred from not only the words a person uses but also *in their description of the situations that have led to the emotion.* This observation aligns with the appraisal theory discussed earlier and also suggests that a computational model of emotion detection could use a transcript of naturally occurring conversation to capture the structure of the chain of actions leading to an emotion Balahur et al. [2011]. The second conclusion is that people can use emotional words to do things other than describe emotion, like supporting or undermining the ”sensibility” of their (or another) person’s actions. This suggests that a computational model of emotion, to be completely accurate, would have to capture the dynamic range of rhetorical techniques and narrative sequences afforded to the speaker. To our knowledge, no such computational model yet exists, although researchers in knowledge-based computing attempt to do so: see for example, Balahur et al. [2011], Poria et al. [2014]. Most of the other models in this review are therefore limited in the range and resolution of emotions that they can accurately detect, typically to: varying degrees of positivity e.g., Curry et al. [2018], Sun et al. [2019] or to a subset of Ekman’s basic emotions (i.e., happiness, sadness, fear, anger, surprise, and disgust) e.g., IBM [2019], Fraser et al. [2018], Mazzoleni et al. [2017]. In the models we review, a fully incremental model of emotion detection remains a challenge.

While it remains a challenging and interesting research problem to recognize a large variety of emotions and to do so incrementally, it may not always be necessary. For example, in call centers it may be sufficient to recognize negative and non-negative emotions to improve the quality of services. Lee et al. [2005] describes a computational model in which discourse is combined with lexical and acoustic spoken language information to infer positive or negative emotion for a call center application. By constraining the domain to call-center dialog and fusing several spoken language features, the researchers showed improved emotion recognition for this specific domain.

In summary, there are subtle but distinct differences between how human express emotions depending whether they use text messages or inter-person, natural conversation. When conversing in person, emotions may be revealed incrementally, evolving from a course of actions. In the general case, computational models will have a difficult time detecting entire range of evolving emotions. However, most models to not aspire to such a general level of performance and the agent typically limits the detected emotions to a set of categories (e.g., Ekman's six emotions).

2.4 Extracting Sentiment from Features

In the following sections, we summarize prior research in computational models of sentiment analysis and organize it by the assumptions made with respect to the sentiment-bearing features in the natural language. Some approaches use features from multiple categories; we point these out.

2.4.1 Bag-of-Words Models

In this section, we review investigations that make the assumption that words contain sentiment-bearing information. In this category are models in which the features are unigram, bi-gram or tri-gram terms. The characteristic that all models in this category share is that the terms are independent from one another; this is called a *bag-of-words* (BOW) model. More generally, n-gram models assume context or grammatical structure are important characteristics and can be found in the syntax, ontological, and language models categories.

Amolik et al. [2016] describe a model in which the features are Twitter keywords (unigram approach). Input is presented to a Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers which predict three classes: pos, negative, and neutral. The class labels were manually entered. The model was trained on 1,800 tweets (600/label). Extracted features are individual words, after removing stop words, punctuations, and repeated chars. The researchers concluded that the SVM classifier was more accurate than NB and claim 75% and 65% accuracy respectively.

Lee et al. [2005] explore the detection of domain-specific emotions using language and discourse information in conjunction with acoustic correlates of emotion in speech signals. The experimental design objective is to detect negative and non-negative emotions using spoken language data obtained from a call center application. Most previous in emotion recognition have used only the acoustic information contained in speech. In their investigation, a combination of three sources of information: acoustic, lexical, and discourse is used for emotion recognition. In their model, acoustic and discourse features are computed separately. Three classifiers each take the acoustic, language, and discourse features and their output predictions are averaged. Admittedly, their input domain is highly constrained and they assumed no ASR transcription errors, which is highly unlikely in practice. Under these conditions, the researcher reported that significant improvements can be made by combining these information sources in the same framework.

In [Pang et al., 2002], the researchers try to gain a better understanding of how difficult sentiment classification is by investigating whether common machine learning (ML) techniques outperform human evaluations. They found that for the problem of classifying a document by overall sentiment, standard ML techniques (SVM, Naive Bayes, Maximum Entropy) outperformed human baselines, but do not perform as well as topic based categorization.

2.4.2 Sentiment Lexicons

A *sentiment lexicon*, is a database of lexical units for a language along with their sentiment orientations. The orientations may be expressed as a set of tuples, e.g., (lexical unit, sentiment). The lexical units may be represented as words, word senses, phrases, etc.. Sentiment, on the other hand, may be represented in several ways, for example: fixed categorization into *positive, negative*, a finite number of graded sets, e.g., strongly positive, mildly positive, neutral, mildly negative, strongly negative, or a real value representing emotional valence in an interval such as $[-1, 1]$ [Ahire, 2014].

Sentiment lexicons associate words in a document with a sentiment "score" [Hutto and Gilbert, 2014, Svetlana et al., 2014, Tausczik and Pennebaker, 2010a]. Balahur et al. [2011] analyzes the syntactic structure of the text and consults domain-specific data banks to capture and store the structure and semantics of events in the text, and use it to predict the emotional responses triggered by a chain of actions. In either of these approaches, language-specific knowledge and hand-crafted lexicons are required for these techniques to be successful. This approach has limitations similar to that of the statistical approaches: different types of text (e.g., blogs, newspaper articles, movie reviews, tweets) require specialized methods [Balahur, 2013, Pang et al., 2008] and cannot be generalized. Finally, some approaches to sentiment analysis combines sentiment lexicon and statistical approaches. Pang et al. [2002] train several classifiers using features extracted from the text using natural language processing tools; additional examples of the hybrid approach can be found in [Calvo and D'Mello, 2010].

The lexicons may be hand-crafted using a dictionary or a corpus. Multiple annotators are used and the inter-annotator agreement is calculated. The major advantage of this approach is that human evaluators use their innate judgment to label the data. From all the annotations for a give unit, statistics can calculated which is helpful in identifying ambiguity (barring human error). Linguistic Inquiry and Word Count (LIWC) [Pennebaker and Francis, 1996, Tausczik and Pennebaker, 2010a, Pennebaker et al., 2015a] is a fee-based textual analysis tool that uses human "judges" to evaluate sentiment. It does much more than provide sentiment scores and its creators psychometrically validated the LIWC dictionaries to ensure that values across LIWC categories have been shown to correlate with big-five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism.

VADER [Hutto and Gilbert, 2014] is the sentiment analysis function in the natural language processing toolkit (NLTK) for the Python programming language. VADER is a sentiment lexicon tuned to microblog-like contexts sensitive to polarity and sensitivity. The lexical features are combined with five syntactical and grammatical rules that embody how humans express intensity. The researchers report that its sentiment lexicon is gold-standard quality as it has been validated by humans. They distinguish it from LIWC in that they report improved performance in social media contexts and that is an open-source tool. The researchers evaluated Vader against several other sentiment lexicon models (e.g., LIWC, GI, ANEW; see below) using standard precision, recall and F_1 metrics on four different test sets: social media, amazon product reviews, Pang et al. [2002] movie reviews, and NY Times editorials. Vader performed better than all other methods in every test domain.

The disadvantage of the manually created lexicons is the sheer size of a typical language corpus; the Oxford English Dictionary, 2^{nd} Edition contains over 290,000 entries [Oxford University Press, 2020]. In response, automatic methods to create sentiment lexicons have been developed. The typical approach is to create a set of starting seed words with known sentiment orientation, and then expand that seed set using an already existing lexical resource. One such freely-available approach is SentiWordNet [Esuli and Sebastiani, 2006], an open-source lexical resource for opinion mining. It assigns to each WordNet synset three sentiment scores: positivity, negativity, objectivity.

There is a trade-off in accuracy between hand-crafted approach and the automatic approach. Striking a balance between the two is an active area of research. In one such approach, Hamilton et al. [2016] combine domain-specific word embeddings with a label propagation framework to induce accurate domain-specific sentiment lexicons using small sets of seed words. The researchers report state-of-the-art performance and that their purely corpus-based approach outperforms methods that rely on hand-curated resources such as WordNet.

Further more, sentiment lexicon performance can be improved by understanding deeper lexical properties such as parts-of-speech; this provides more context awareness. A lexicon can be fine-tuned using the process of word-sense disambiguation (WSD), which identifies which sense of the word is being used. The VADER researchers state that it is a goal to provide such a fine-tuned lexicon.

Other sentiment lexicons that have been widely-used include: Harvard's General Inquirer (GI), designed as a tool for content analysis, providing sentiment polarity; Hu and Liu [2004] made publicly available a binary polarity sentiment lexicon of nearly 6,800 words; Affective Norms for English Words (ANEW) a valence-based lexicon [Bradley and Lang, 1999]; SenticNet, a publicly available semantic and affective valence-based lexicon for concept-level opinion and sentiment analysis [Cambria et al., 2012a]. A detailed discussion of these is contained in [Hutto and Gilbert, 2014, 217]. Kiritchenko et al. [2014] describes a sentiment analysis system which uses generally-available, manually created sentiment lexicons combined with automatically generated social media-specific sentiment lexicons. These were used to detect binary emotion in short, informal texts.

2.4.3 Topics as Features

Topic modeling use a bag-of-words approach to discover the topics contained in a collection of documents. The idea is that the topic distribution in a document can be correlated with emotion. For example, the extent to which one talks about different topics (i.e., the topic proportions) when recounting a pleasurable experience is likely to be significantly different than when talking about an unpleasant experience. Thus we can infer emotion from the topics we talk about. This, in theory, might avoid some of the challenges using language features whose sentiment polarity changes depending on context. We discuss topic modeling in detail in Chapter 3, Section 3.2. In this section, we review other models which applied topic modeling to extract sentiment.

Shah et al. [2013], describe a speech-based emotion recognition framework based on Latent Dirichlet allocation (LDA). The system finds topics using incoming speech frames rather than from the text and uses an SVM classifier to identify seven emotions. Their model used test data from EMO_DB, in which actors spoke highly-exaggerated emotions. For this non-naturalistic experiment, the researchers report their model achieves a classification accuracy of 80.7%. In an extension of this work, Shah et al. [2015] use a novel acoustic feature extraction approach in which a supervised replicated softmax model is proposed to learn naturally discriminative topics. Topic are then used to train a classifier for emotion recognition of four categories: sad, happy, angry, and neutral. The researchers report a 16.75% improvement over other methods.

Lin and He [2009] propose a novel probabilistic modeling framework based on LDA, called joint sentiment/topic model (JST), which detects sentiment and topics simultaneously from text. Unlike other approaches which often require labeled corpora for classifier training, the proposed JST model is fully unsupervised. The model has been evaluated on the Pang et al. [2002] movie review dataset to classify the review sentiment polarity. To achieve performance close to 2009 ”state-of-the-art” (i.e. BOW/SVM, 90%), requires incorporating prior information. While the authors insist this does not violate, their ”unsupervised” model categorization, it does require selecting sentiment-bearing words (”paradigm words”).

2.4.4 Using Semantics

In this section, we review approaches that assume sentiment can be correlated with the context or meaning of words. This is done by encoding a word as a vector in a process called word embeddings. These dense vectors generalize better and tend to do a better job of capturing synonymy then sparse vectors (representing a word as a “hot vector” whose dimension is all the words in the lexicon). In this way, the word vector can capture the similarity between, for example, “car” and “automobile” [Jurafsky, 2000]. One may think of these vectors as representing similarity as distance in semantic vector space.

Mazzoleni et al. [2017] describe a simple way to detect emotional states using word embeddings. Their model assigns the percentage of Ekman’s basic emotions (“anger”, “disgust”, “sadness”, “happiness”, “fear”, “surprise”) to short sentences.

The method has been tested on a collection of Twitter messages and on the SemEval 2007 news headlines dataset. The entire period is expressed as the mean of the word's vectors that compose the phrase, after preprocessing steps. The sentence representation is finally compared with each emotion's word vector, to find the most representative with respect to the sentence's vector. Their method predicts "disgust", "happiness", "sadness", and "surprise" labels with an average $F_1 = 55$; it struggles with "anger" and "fear" as they had no labels in the training set. The method performs better predicting binary polarity: average $F_1 = 77$.

Kim et al. [2010] describe a study which estimates a categorical and dimensional model for the recognition of four affective states: "anger", "fear", "joy", and "sadness" that are common emotions in three datasets: SemEval-2007 "Affective Text", ISEAR (International Survey on Emotion Antecedents and Reactions), and children's fairy tales. In the first model, WordNet-Affect is used as a linguistic lexical resource and three dimensionality reduction techniques are evaluated: Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), and Non-negative Matrix Factorization (NMF). In the second model, ANEW (Affective Norm for English Words), a normative database with affective terms, is employed. Experiments show that a categorical model using NMF results in better performances for SemEval and fairy tales, whereas a dimensional model performs better with ISEAR.

Balahur et al. [2011] describe a model that utilizes semantics, semantic lexicons, syntax, and an ontology to detect emotion in text. What makes their approach stand out is the observation that detecting emotion in text is hard because expression of affect results from interpretation of meaning and from context as it interacts with the text. Systems that operate that detect emotion at the word level and cannot express what a human would perceive as emotion. Their model, EmotiNet, provides a way to capture and store the structure and semantics of events and predict the emotional responses triggered by a chain of actions. It is based on the Appraisal psychological theory of emotion, discussed in Section 2.2.

The model was evaluated using real life situations drawn from the "family situations" domain of the ISEAR data bank. Performance was better than chance and the model is more flexible than systems which are sensitive to the vocabulary they are trained on. However, performance of the model is limited by the quantity of knowledge stored in the knowledge base, which requires further extension in order to increase the precision of emotion classification and its successful applicability to domains other than family situations.

Strapparava and Mihalcea [2008] describes experiments to automatically analyze emotions in text. The researchers discuss construction of a large data set for Ekman's six basic emotions. They compared Sentiment and Semantic approaches to a baseline Naive Bayes classifier. For the Sentiment Lexicon, WordNet-Affect was used. For the semantic model: LSA; LSA augmented with words from WordNet synset; LSA augmented with WordNet synset and sentiment labels from WordNet-Affect. The evaluation was carried out on a dataset developed for SemEval 2007 (news headlines). The researchers reported best results using LSA + WordNet-Affect: $F_1 = 17.57$ (recall: 90.22; precision: 9.77; this suggests the model has a high rate of false positives).

Turney [2002] discusses a simple unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. The semantic orientation of a phrase is calculated as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". A review is classified as recommended if the average semantic orientation of its phrases is positive. The algorithm achieves an average accuracy of 74% when evaluated on 410 reviews from Epinions, sampled from four different domains (reviews of automobiles, banks, movies, and travel destinations). The accuracy ranges from 84% for automobile reviews to 66% for movie reviews.

2.4.5 Sentiment from Syntactic Structure

As demonstrated by Strapparava and Mihalcea [2008], unsupervised, semantic approaches alone usually perform no better than supervised bag-of-words models depending on the test database, but when augmented with additional features, show improvement. However, neither of these approaches can express the meaning of longer phrases properly, without using a compositional approach. In this section, we review models that use properties of the language grammar to infer sentiment. On one hand, these approaches are more language-specific, but this allows a model to use syntactic properties to detect sentiment of an aspect of a product, for example.

Socher et al. [2013] introduce a *Sentiment Treebank*. It includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences. The model is a Recursive Neural Tensor Network (RNTN) trained on the Treebank. It is a deep neural network (DNN) that takes a phrase of any length and represents it as word vectors and a parse tree. It computes vectors for higher nodes in the tree using same tensor-based composition function. The researchers report that the model pushes the state-of-the-art performance (as of 2013) in single sentence positive and negative classification from 80% up to 85.4%. Fine-grained, 5-class sentiment labels for all phrases reaches 80.7% accuracy, an improvement of 9.7% over BOW baselines. The researchers claim that it is the only model that can accurately capture the effects of negation and its scope at various tree levels for both positive and negative phrases. A demo and the source code for the Sentiment Treebank and model is available at [Chuang et al., 2013].

Kiritchenko et al. [2014] describe their model which participated in the SemEval-2014 Task 4 competition on aspect-level sentiment analysis. The competition challenges were to detect: aspect categories, sentiment towards aspect categories, aspect terms, and to detect sentiment towards aspect terms in the laptop and restaurant domains, respectively. The researchers approached this as a “feature engineering” problem, hand-crafting word, lexicon and syntactic features specifically for this challenge. For word features, unigrams, bigrams were used. For lexicon features, they augmented the supplied training and development dataset with a large corpora of restaurant and laptop reviews. These were used to create: sentiment lexicons for laptops and restaurants; word-aspect association to identify aspect categories of restaurants; word-clusters were generated from the restaurant reviews and publicly available tweets. For syntactic features they used: part-of-speech tags, and for context, cluster ngrams. An SVM classifier was used to discriminate Aspect Term Polarity (pos, neg, neu, conflict), Aspect Category Detection (food, price, service, ambiance, misc) and Aspect Category Polarity (pos, neg).

The evaluation used standard machine learning metrics (accuracy, precision, recall, F_1) and across the subtasks, their lowest score was $F_1 : 0.68$ and highest, $F_1 : 0.88$. However, the point of the challenge is to see how the author's submission compared to the other thirty. In that, their submissions stood first on 3 out of 4 subtasks, and within the top 3 best results on all 6 task-domain evaluations. This demonstrates the importance feature engineering for the benefit of solving a specific sentiment analysis task.

Agrawal and An [2012] used semantic and syntactic relationships to detect Ekman's six basic emotions. Unlike the approach of Kiritchenko et al. [2014], the investigators do not rely on a hand-crafted lexicons which they claim give the ability to generalize beyond the training data set. Their basic approach is as follows. In the pre-processing stage, sentences are parsed for POS-tagging and syntactic dependencies to identify context (e.g., adj complements: looks, beautiful; adjective modifiers: meat,red; and negation: happy,not). Semantic-relatedness computes emotion vector of affect-bearing words by calculating relatedness to emotion concepts. In the syntactic stage, phrase-level analysis uses context to adjust the emotion vectors. Sentence analysis aggregates emotion vectors to label the sentence's emotion.

The model was compared against: Keywords(WordNet-Affect), other semantic models: Latent Semantic Analysis (LSA), Probabilistic LSA, Negative Matrix Factorization (NMF). It was tested on ALM (Fairy-tales) dataset in which four classes were predicted: "Happy", "Sad", "Anger-disgust", "Fear". The average F_1: 0.61. ISEAR (persons with different cultural backgrounds asked about their emotional experiences) which has seven emotional labels but only four used: “Joy”, “Sad”, “Ang-Dis”, “Fear”. The average F_1: 0.55. ISEAR - all 7 emotions (“Joy”, “Sad”, “Anger, “Fear”, “Disgust”, “Shame”, “Guilt + Average”). The average F_1: 0.43. This model performed better than the semantic approaches it was compared against, thus providing some insight as to the level of performance that can be expected in an unsupervised approach which augments semantic-relatedness with context derived from analyzing syntactic-dependencies.

The IBM Tone Analyzer [IBM, 2019], a fee-based service, derives emotion scores from text, using a stacked generalization-based ensemble framework; stacked generalization uses a high-level model to combine lower-level models to achieve greater predictive accuracy. Features such as n-grams (unigrams, bigrams, and trigrams), punctuation, emoticons, profanity, greetings (such as “hello”, “hi” and “thanks”), and sentiment polarity are fed into machine-learning algorithms to classify four emotion categories: '“anger”, “fear”, “joy”, “sadness”.

The training set was labeled in a human evaluation of 200,000 sentences culled from debate forums, speeches, and social media. The model was evaluated using the ISEAR dataset with a reported avg F_1 : 0.41 against a 0.37 claimed prior state-of-the-art. It was also evaluated using SEMEVAL with a reported avg F_1 : 0.68 against a 0.63 claimed prior state-of-the-art.

2.4.6 “Common Sense” (Sentic) Computing

In this section, we review the concept of *sentic computing.* The concept of Sentic computing is discussed in detail by Cambria and Hussain [2012] in their book *Sentic Computing: Techniques, Tools, and Applications.* It relies on the ensemble application of “common sense” computing and the psychology of emotions to infer the affective information associated with natural language. What led to sentic computing is the need for better accuracy when switching between domains. The idea is to use concepts to allow the system to perform opinion mining across domains. Key to its processing are the linguistic dictionaries which are used to interpret emotion-bearing indicators in the text. The processing also uses a parser which deconstructs text into concepts using a lexicon based on concepts extracted from knowledge databases; these are then fed into a vector space of common-sense knowledge, called AffectiveSpace. Concepts are used to find similarity with the knowledge already stored in AffectiveSpace. The AffectiveSpace is clustered on the Hourglass psychological model of emotion [Cambria et al., 2012b], inspired by Plutchik. The idea is to reason on the semantic and affective-relatedness of natural language concepts in the input and in the AffectiveSpace and use this infer emotion.

Poria et al. [2014] describes these concepts in detail and describes an extension to improve Cambria's original model. This model adds discourse patterns to allow the sentiment to flow from concept to concept based on the dependencies in the input text to gain a better understanding of the conceptual role of each concept. Using this, the model generate an emotional valence polarity based on the speaker's feeling. The model was tested on a movie review data base of Pang et al. [2002] and the researchers reported emotion polarity prediction accuracy results that exceeded the then state of the art (by 0.8%) reported by Socher et al. [2013]. The model was also tested on a corpus of product reviews from seven other domains and achieved an accuracy of 87%, although a reference accuracy does not exist for comparison. Precision, recall, and F_1 classification metrics were not given.

From this experiment, sentic computing, achieves results superior to BOW models at least when detecting emotional polarity (i.e., positive, negative). It achieves this at the cost of utilizing rich knowledge databases which represent an aggregation of conceptual and affective information available from the Web. Given the claimed richness of the model and the fact that uses a categorical and dimensional psychological emotion theory, we expected an evaluation of the model's ability to predict fine-grained emotions, as Socher et al. [2013] did. In addition, more challenging sentiment analysis problems, such as aspect and category detection, are not explored by this model.

2.4.7 Pre-trained Language Models

We saw in the previous section, an example in which feature engineering can significantly improve results is a specific language task: aspect-based sentiment analysis. The goal of pre-trained language models is to automatically discover the features that are needed to solve a language problem and, hopefully, push the state-of-the-art performance levels while doing so. More formally, given a context, a language model predicts the probability of a word occurring in that context. They do so by capturing deep contextualized word representations that captures syntax and semantics and how these uses vary across language contexts. This is effective because this method forces the model to learn how to use information from the entire sentence to infer the missing words.

Although there are several examples of such models under investigation, we will look at two of the most notable examples. The first is ELMo (Embeddings from Language Models) [Peters et al., 2018]. The word embeddings learned are from a large text corpus and then applied to a number of existing models for language tasks. Of particular interest to us is the result when ELMo was applied to the fine-grained sentiment analysis task in the Stanford Sentiment Treebank that was described in Section 2.4.4. ELMo word embeddings were applied to the biattentive classification network (BCN) used in [McCann et al., 2017], which previously achieved state-of-the-art-performance. The ELMo word embeddings replaced the original input embeddings to BCN and achieved a 1% increase performance for the fine-grained sentiment analysis task.

The second model, BERT (Bidirectional Encoder Representations from Transformers), improves on ELMo by training deep bidirectional representations from unlabeled text by jointly conditioning on both right and next context in all layers of the network [Devlin et al., 2018]. In other words, it is deeply bidirectional, as opposed to ELMo which uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for the downstream task. The approach to using BERT for a particular language task is to “fine-tune” the model by swapping out the appropriate inputs and outputs; the general pattern for doing so is described in the paper. Sun et al. [2019] describe fine-tuning BERT for aspect-based sentiment analysis. After doing so, the authors report performance accuracy significantly exceeding benchmark methods used in SemEval-2014 paper; their source code, as well as BERT and ELMo, is freely-available.

The paper that introduced BERT was published in 2018, as was ELMo. The results reported so far in the research community have been highly-encouraging as the models seem to be setting new performance benchmarks in natural language processing. Anecdotally, BERT bests (acc. 94%) the performance of two models: one that uses word embeddings and POS tags as features to a logistic regressions classifier (acc. 91%) and one that uses a BOW model whose terms are weighted by their importance as input to a convolutional neural network,(acc. 85%). On the other hand, little has been published about which tasks these models are particularly good at and what representations they are learning to make this so.

2.4.8 Using Audio Features in the Speech Signal

Although the primary focus of this chapter is detecting emotion in conversational text, we now explore the extent to which speech transmits affective information from the paralinguistic features of speech (i.e., how it is being said). Humans seem to be able to infer basic emotions from prosody and non-linguistic vocalization (e.g., cries or laughs) [Juslin and Scherer, 2005]. Calvo and D'Mello [2010, Table 2] compare selected investigations where voice was used to recognized affect and we highlight some of the findings as follows. Schuller et al. [2005] describe an approach where acoustic and linguistic features are fused and presented as input to an ensemble classifier. While the authors demonstrate that fusion approach offers improved performance, they show this only when the acoustic features are speaker-dependent. Furthermore, the researchers used "acted emotions" when extracting the acoustic features, which may have exaggerated the affect in the voice signal, making it easier to detect.

Acoustic features are also used to detect emotion; research has shown that parameters such as pitch, intensity, rate of speech and voice quality are important features in detection emotion [Murray and Arnott, 1993]. Lee et al. [2005] fuse acoustic correlates of speech, lexical, and discourse data to predict positive and negative emotion in call center dialogues. They measured classification accuracy and found that best performance was achieved when both acoustic and language sources were combined. Furthermore, performance varied by gender, with classification accuracy higher for those reporting as females. When acoustic-only features were used, accuracy was approximately 10% lower than when language alone was used. This suggests the primacy of the language channel.

Poria et al. [2017] is an excellent review of recent unimodal and multimodal techniques for affective computing. In their review, the researchers point out that the acoustic features that are used to detect emotion are dependent on the personality traits of a person. In addition, the review also supports the results described in Schuller et al. [2005]; the speaker dependent approach gives much better results than the speaker independent approach. However, the researchers report that the best accuracy achieved for speaker-independent emotion detection using acoustic features is approximately 81%; this is about 10-15% lower than benchmarks reported when using text. Human ability to recognize emotion in speech audio is approximately 60%; sadness and anger are more easily detected than joy and fear. [Scherer, 1996].

Thus, even though the computational approach appears to outperform humans, the results also suggest that acoustic features alone is insufficient to accurately detect emotion in conversational language. This is consistent with the findings described by Krauss et al. [1981] who suggest that there is no support for the assumption that nonverbal communications for the primary basis for observing affect.

2.5 Challenges for Sentiment Analysis

There are several areas which can confound a sentiment analysis system and we present the main one as follows. For further a comprehensive review, Mohammad [2017] discusses several problems that are the focus of recent investigations in sentiment analysis.

2.5.1 Language Structure

The prior polarity of a word can be used to deter whether a word conveys positive or negative emotion be analyzing its prior polarity. However, the prior polarity can change depending on the context in which the word appears. Further, negative words can be used in phrases that express positive valence and vice versa. Detecting negation in phrases is not always syntactically easy. There may be phrases in which one word in the phrase is positive and at least one word is negative (e.g., "happy accident"). Negation of positive words often make them negative, yet make negative words less negative (not positive). There are subtle aspects of negation that are still under investigation. For example: how to people in different cultures use negation differently; to what extent does a given negator impact sentiment more and which ones less? Other areas that remain unexplored include investigating how degree adverbs (barely, moderately, slightly) and intensifiers (too, very) impact sentiment of the predicate [Kiritchenko and Mohammad, 2017].

2.5.2 Reliability of the Ground Truth

Social media, movie, and product reviews often contain sentiment indicators such as the number of stars, emoticons; for these there are labeled datasets, e.g., [Pang and Lee, 2012]. The ground truth is not always readily available for sentiment occurring in natural language, and as a result the approach is to "crowdsource" the evaluation, (see Chapter 4 Section 4). However, two people, reading the same text in different contexts will come to different conclusions about sentiment, especially on borderline cases. Thus it is important for sentiment analysis researchers to report in detail what measures were used to determine and ensure reliability of the ground truth, through agreement of metrics, for example.

2.5.3 Speech Transcription Errors

Transcription errors from the ASR can affect the accuracy of the sentiment analysis computational model. State of the art ASR systems typically have a WER of about 15% which is still means one word our of eight will be either deleted, substituted, or superfluously added to the transcription. In noisy environments the error rates can be significantly higher, thus improvement of sentiment classification on erroneous ASR transcription is necessary for a practical intelligent agent. There few approaches that are discussed in the literature. Dumpala et al. [2018] discusses system that combines transcripts with audio and visual modalities available during training to improve the ASR transcription during testing. Chen et al. [2017] discusses how to reduce ASR errors in a chatbox using sequence-to-sequence model. Sheikh et al. [2017] discusses the use of named entity recognition to improve speech recognition when its input contains out-of-vocabulary words, most of which are proper nouns.

2.5.4 Domain Generalization

Sentiment analysis models can be sensitive to the domain they were trained on. This implies the need for a large amount of training data for a multi-domain model. One reason that there can be words that have different sentiment polarities depending on the domain. Cai and Wan [2019] gives an example in which the phrase “It is easy” denotes positive sentiment and when used in the domain of movie reviews, it is negative: “The end of the movie was easy to see coming”. The researchers leverage the attention mechanism described in [Vaswani et al., 2017] which allows models to focus on the more important words and phrases. The goal of Cai et al. is to pay more attention to significant words in a text and identify their sentiment polarities in each specific domain. Their work builds upon Multi-task learning [Caruana, 1997] which has been used to solve multi-domain sentiment classification.

2.6 Summary

In this chapter, we discussed the primary models of psychological emotion that inspire many of the computational models of sentiment analysis. We then reviewed representative computational models, organized by the assumptions they make about the emotion-bearing features of the text: bag of words, sentiment lexicons, topic, and semantics. We also reviewed examples from Sentic Computing that use a richer, knowledge-based approach. Finally, we discussed two influential pre-trained language models, ELMo and BERT which are at the forefront of setting new performance benchmarks in language processing. We also showed how BERT was applied to aspect-detection.

We also reviewed the contribution of using audio and prosodic features available in the speech signal. When the audio channel is available, recent research suggests that it can improve the accuracy of emotion detection. However, maximum performance is achieved when the model is trained on an individual speaker's audio characteristics. For a general population conversational agent, this of course is impractical. However, for an agent situated in a companion robot, individualization to the person's speech characteristics is entirely possible.

We discussed the main challenges in sentiment analysis. The ground truth in conversational text is hard to come by; thus nearly every model has been trained and tested on heavily opinion-bearing text (e.g., movie reviews, product reviews, news headlines, social media). Reliably labeling conversational text is difficult because several humans may come to different conclusions given the same text to analyze. Thus ensuring inter-rater reliability is important. Alternatively, collecting *all* the ratings, reliable or not, and measuring the model's deviation from the rater mean might be a better way to evaluate model performance.

Natural language is context sensitive and sentiment often arises from the context in which it is situated. We noted better performance in models that take context into account, such as the syntactic models which were state-of-the-art in 2013 until the pre-trained language models arrived on the scene in 2018.

We observe that most of the peer-reviewed models were not developed to analyze conversational text; they are trained and tested on data sets that have a great deal of sentiment-bearing text. This presents a challenge when building a companion robot or a spoken dialog system. On the other hand, the IBM Tone Analyzer, for which there are no peer-reviewed publication, is an exception. It is designed for conversation applications such as a call-center. Fraser et al. [2018] used the IBM Tone Analyzer to detect "joy", "anger", and "sadness" as a user interacted with an NPC (non-player character) in a video game. The idea was to see if a conversational AI systems without emotional dialogue leads to less engagement (at least in a role-playing game). The researchers found that users spent longer in the conversation when using the emotional version.

We also note the existence of the Alexa Challenge competition [Ram et al., 2018]. In this competition, participants are to design a dialogue system that engages the users for as long a period as possible. Some competitors included an emotional score to help evaluate the user utterance and the system's response. The dialogue system described by Curry et al. [2018] used the ratio of users' turns containing some predefined key phrases such as"that's pretty cool", "you're funny", "gee thanks" or "awful", "you're dumb" together with the sentiment polarity of those turns as the approximation of user feedback and engagement. The system used VADER to infer a continuous level of positivity in the user's utterance. Their emotional dialogue system achieved consistently high user ratings and long conversations throughout the semifinals period of the competition.

Chapter 3

Inferring Emotional State in Persons with Parkinson's Disease

Individuals with Parkinson's Disease (PD) often exhibit facial masking (hypomimia), which causes reduced facial expressiveness. This can make it difficult for those who interact with the person to correctly read their mental state and can lead to problematic social and therapeutic interactions. In this chapter, we develop a probabilistic model for an assistive device which can automatically infer the mental state of an individual with PD using the topics that arise during the course of a conversation. We envision that the model can be situated in a device that could monitor the emotional content of the interaction between the caregiver and the person living with PD, providing feedback to the caregiver in order to correct their immediate and perhaps incorrect impressions arising from a reliance on facial expressions. We compare and contrast two approaches: using the Latent Dirichlet Allocation (LDA) generative model as the basis for an unsupervised learning tool, and using a human crafted sentiment analysis tool, the Linguistic Inquiry and Word Count (LIWC). We evaluated both approaches using standard machine learning performance metrics such as precision, recall, and F_1 scores (i.e., harmonic mean of precision and recall scores). Our performance analysis of the two approaches suggests that LDA is a suitable classifier when the word count in a document is approximately that of the average sentence, i.e., 13 words. In that case the LDA model correctly predicts the interview category 86% of the time and LIWC correctly predicts it 29% of the time. On the other hand, when tested with interviews with an average word count of 303 words, the LDA model correctly predicts the interview category 56% of the time and LIWC, 74% of the time. Advantages and disadvantages of the two approaches are discussed.

3.1 Introduction

Parkinson's disease (PD) is a universal disorder with an incidence ranging from 9.7 to 13.8 per 100,000 population per year [WHO, 2006]. In the US, PD follows Alzheimer's disease as the most common neurodegenerative disorder, affecting at least 500,000 Americans and perhaps 500,000 more if we include the undiagnosed and misdiagnosed cases [NIH, 2018]. Tremors, muscle rigidity, bradykinesia (slowness of movement), and loss of balance are symptoms which accompany the disease; it is progressive, the symptoms worsening over time. The first three symptoms can occur in the facial, respiratory, and vocal muscles, resulting in diminished control of one's facial and vocal expression which can dissociate one's inner emotional state from the outward facial appearance; this is known as facial masking and is called hypomimia. Because people rely heavily on facial expression in attributing and interpreting other's emotions and motivational states, facial masking can deeply affect the person's ability to communicate which may lead to impaired social interactions and reduced quality of life [Sturkenboom et al., 2013, Takahashi et al., 2010]. For example, rehabilitation therapists often use a client's verbal and nonverbal behavior to infer the client's emotional state; if the client is mostly silent or displaying little facial expression, the therapist may infer the client to be more hopeless or apathetic which may not be their true emotional state. In the home and community, desynchronization between a person's emotional state and her external expression can occur during any social situation, which might take place in the home, among family and friends, and at work [Takahashi et al., 2010]. This may exacerbate feelings of social incapacitation and stigmatization, which leads to reduced quality of life and the vicious cycle of decreasing social engagement [Ma et al., 2016].

Given that facial expressiveness is a problematic channel for communicating emotions and emotional states in people with PD, a more accurate channel might be verbal communication: the words a person uses in their verbal or written speech [De-Groat et al., 2006]. Since for humans it is very difficult to override the interpretation of information transmitted through facial expression, which happens automatically and instinctively, it would be helpful to have a reliable, automated way of analyzing verbal communication that helps detect the valence of the emotion expressed. This automated capability could be implemented in a communication-assistive tool for improving social life. The tool could take the form of a robotic companion or an application that would help people living with PD, their caregivers, and social community by alerting conversation partners to misunderstandings coming from the desynchronization the person with PD experience of emotion and its reflection in the face. This device is meant to improve natural human interaction in the home and community.

For text, detection of emotional content and its valence has been attempted using an automated textual analysis software program called Linguistic Inquiry and Word Count (LIWC) [Pennebaker et al., 2015b]. Tausczik and Pennebaker [2010b] showed that LIWC's categories for positive emotion, negative emotion, anxiety/fear, anger, and sadness/depression were correlated with external raters' judgments, demonstrating they can be used to assess emotional content in text. However, there are several limitations to using the LIWC approach. The basis for LIWC's text analysis is a dictionary which in the latest version (i.e., LIWC2015) consists of approximately 6,400 words, word-stems, and emoticons, i.e., a pictorial representation of human facial expressions used to convey emotion in text [Pennebaker et al., 2015b]. LIWC's dictionaries are constructed by human scientists according to evaluation data generated by human raters, rather than learned from the text automatically. This means that LIWC cannot be used with natural languages for which the software has not been modified to accommodate (i.e., for which such dictionaries have not been created). LIWC relies on word recognition, and needs to be periodically updated as language usage evolves. Also, LIWC is not designed for spoken language [Pennebaker et al., 2015b], while for the detection of emotional content in a conversation, the ability to work with spoken language is crucial.

In this paper, we introduce a novel approach: using the Latent Dirichlet Allocation (LDA) generative model as the basis for an unsupervised learning tool which is trained to extract topic proportions from a collection of text documents (see Sections 3.2 and 3.3 for details). When an unseen document is presented to the model, it finds the document's topic proportions and uses them as a set of features. We then use a logistic regression (LR) classifier to associate these features with training data having enjoyable emotional content (text obtained through the prompt: talk about an enjoyable experience) or frustrating emotional content (text obtained through the prompt: talk about a negative emotional experience). We compare our model with the LIWC approach: the word count frequency of five LIWC features associated with emotion is extracted from the text and these are used to train another LR classifier to associate them with the emotion content labels frustrating and enjoyable that have opposing valence.

The paper proceeds in the following way. In *Background and Related Work*, Section 3.2, we review the LIWC and LDA approaches. In Methods, Section 3.3, we show how interview transcripts from the *Self-management Rehabilitation and Health-Related Quality of Life in Parkinson's disease* database [Tickle-Degnen et al., 2010] were used as text documents to train and test both models and we compare the results of two experiments: the first using training and test documents from the entire interview (average word count = 303) and the second using documents which were edited to contain the first 20 seconds of the interview transcripts (average word count = 13). The Results section shows that for longer text, the LDA model correctly predicts the emotion label (frustrating or enjoyable) 56% of the time while the LIWC model 74% of the time. However, in the case of shorter text, the LDA model outperforms the LIWC model. We then discuss advantages and disadvantages of each approach and potential ways of using them to create emotion detecting assistive conversation tools.

3.2 Background and Related Work

3.2.1 Human-curated approach: LIWC

A reliable method for analyzing the emotional content of text is useful in a wide range of scenarios such as opinion mining where it is necessary to detect shifts in customer sentiment as expressed in social media. One approach is to manually label words according to their semantic valence, either positive or negative [Liu, 2010], creating a sentiment lexicon. Generating a reliable sentiment lexicon manually is time-consuming and thus most researchers rely on already-generated lexicons such as LIWC [Hutto and Gilbert, 2014]. LIWC was first introduced in 1993 and has been updated three times since; its latest version was released in 2015. As previously indicated, LIWC2015 contains an internal default dictionary that is used to determine the words which should be counted in the documents. The dictionary of approximately 6,400 words is associated with particular domains, such as negative emotion, and these are called word categories. There are 41 word categories associated with a psychological category (e.g., affect, biological processes), six personal concern categories (e.g., home, work, leisure), five informal language markers (e.g., swear words, net-speak), and 12 punctuation categories. When a word in the text is found in the dictionary, all the word categories that it belongs to have their counts incremented [Pennebaker et al., 2015b]. The reliability of LIWC has been validated internally (e.g., checking whether the more a person uses a word from a LIWC word category in a text, the more the person uses other words from the same category). The external validity of the LIWC categories have been assessed in contexts relevant to daily living and mental and physical health [Tausczik and Pennebaker, 2010b].

With regards to emotional expression in PD, Takahashi et al. [2010], using data from the same database as this study, measured expressive behavior in transcripts of 212 video clips of 106 persons living with PD by using LIWC to count the number of motivation-related words in each transcript. The videos were recordings of interviews in which the participants were asked to discuss an enjoyable or frustrating activity that occurred during the past seven days. The researchers reported that when participants discussed enjoyable activities, they tended to use more words associated with the LIWC positive emotion category compared to when they discussed frustrating activities. Conversely, participants tended to use more words associated with the LIWC negative emotion category when discussing frustrating activities. The research objective of the current study is to determine whether our machine learning model can achieve similar results to LIWC, using the participants' interview transcriptions from the Takahashi et al. [2010] study and from Tickle-Degnen et al. [2010] study.

3.2.2 Latent Dirichlet Allocation (LDA)

Generating and maintaining a sentiment lexicon suitable for reliably extracting emotional content and its valence from text is a labor and time-intensive undertaking. For example, there have been three major releases since LIWC's initial release in 1993, each containing a new dictionary and improved software design, the result of human testing and validation as well as software engineering effort [Pennebaker et al., 2015b]. To this end, automated approaches to identifying and extracting features from documents which are correlated with emotion valence and intensity have been the subject of active research. We categorize these approaches as machine learning, i.e., the fields of study in which computers learn without explicitly being programmed. In contrast to LIWC in which humans have carefully associated words to emotion categories via its dictionary, the challenge for designing a machine learning model is to identify the features contained in the text, i.e., characteristics of the text that can be used to consistently identify distinctive categories, such as enjoyable vs. frustrating emotional content. The goal is to find features such that as words associated with emotion valence change or new ones are introduced, the model's features also adapt.

Such features can be found in the thematic structure of a document. Topic modeling is the detection of the thematic structure of a document collection; it is a classic problem in natural language processing. One of the motivations for research in this area is to find ways to reduce the dimensionality of large collections of text; the goal is to find semantic structures in the text, which can be used to represent its characteristics using a parsimonious amount of information. This lower-dimensional representation can be used, for example, as an efficient way to retrieve the text.

If samples of text were obtained, we hypothesize that a collection of text documents will contain a mixture of topics. The proportions of these topics in a single document could reflect the enjoyable and frustrating topics contained in that document. Thus, the model design goal is to detect thematic, topic information contained in a sufficiently large sample of text (i.e., a document collection) so that when a document the model has not yet seen is presented, it can identify the proportion of topics contained therein. We then train a classifier to associate a large sample of documents whose emotion valence is already known with these topic proportions. Once that is done, we now have created a way to predict the valence of the emotional content (e.g., enjoyable or frustrating) of any document for which we have extracted its topic proportions. For the feature extraction component of our model design, we will draw from the field of topic modeling, using a technique called Latent Dirichlet Allocation [Blei et al., 2003a].

LDA is built around the intuition that documents exhibit multiple topics. LDA makes the assumption that only a small set of topics are contained in a document and that they use a small set of words frequently. The result is that words are separated according to meaning and documents can be accurately assigned to topics. LDA is a generative data model which as the name implies describes how the data is generated. This idea is to treat the data as observations that arise from a generative, probabilistic process, one that includes hidden variables, which represent the structure we want to find in the data. For our data, the hidden variables represent the thematic structure (i.e., the topics) that we do not have access to in our documents. Simply put, a generative model describes *how* the data is generated, and *inference* is used to backtrack over the generative model to discover the set of hidden variables which best explains how the data was generated. To express the model as a generative probabilistic process, we start by assuming that there is some number of topics that the document contains and each topic is a distribution over terms (words) in the vocabulary. Every topic contains a probability for every word in the vocabulary and each topic is described by a set of words with different probabilities reflecting their membership in the topic. The LDA generative process can be described as follows:

For each document:

1. Choose a distribution (i.e., list of topic proportions) over the topics in the document: $P(\Theta_d)$, which is the per-document topic proportion drawn from a Dirichlet distribution. Note that we have a collection of documents and are choosing a distribution for one of the documents in the collection. The eponymous Dirichlet in Latent Dirichlet Allocation is the name of the distribution that can be used to sample from a collection of distributions.
2. Repeatedly draw a topic from this distribution. Draw a word, w, from the distribution of words for that topic, with the probability $P(w|Z, \beta_k)$, where Z is the hidden topic assignment and β_k is the topic distribution over all the words in the vocabulary. Note that β_k is a Dirichlet distribution as we have a collection of topics from which we are choosing a distribution over words.

For another document repeat (1) and (2). The above process generates each document on a word by word basis, according to the assumptions made about the document's thematic structure (i.e., topic proportions and word distribution), regardless of word order; this latter characteristic is known as a *bag of words* model. We never get to observe this structure, so it must be inferred by asking: (i) what are the topics that generated these documents? (ii) for each document, what is the distribution over the topics associated with that document? (iii) for each word, which topic generated the word? In other words, we want to infer the topic structure which can be thought of, in probabilistic terms, as computing the posterior distribution of our generative model: $P(P_hv|P_o)$ where P_hv is the probability that the document collection has a thematic structure given P_o, the probability of observing the document collection. Operationally, the hidden variables represented by P_hv can be computed several ways using a class of algorithms known as approximate posterior inference. In our model, the LDA algorithm computes both the hidden variables Z (per-word topic assignment) and Θ_d per-document topic proportion). We hypothesize that the topic proportions are features which are reduced-dimensionality representations of the original documents and preserve essential characteristics such as the valence of the emotional content of the text. Our model uses a machine learning classifier to systematically correlate these features with PD participants' interviews, labeled according to their enjoyable or frustrating emotional content.

3.3 Methods

3.3.1 Materials

Input to our model is a document collection of de-identified transcribed interviews collected during a previously conducted randomized control trial called *Self-management Rehabilitation and Health-Related Quality of Life in Parkinson's disease* [Tickle-Degnen et al., 2010]. Data for the current study include responses to open-ended questions about daily life events in the recent past that participants had experienced as particularly frustrating or enjoyable. Participants (N = 117) were people in the early to middle stages of PD, with mild unilateral or bilateral symptoms, Hoehn & Yahr stages 1 through 3 [Goetz et al., 2004], were unassisted for walking and communicating, non-depressed, and of normal mental status. Of the 117 participants, 69.8% were male and 30.2% were female with an average age of 65.6; on average, participants were diagnosed with PD seven years prior to the study. At the time of the interview, participants were "on stage" (i.e., they were taking their medication and their medication was working).

Using a mood-manipulation protocol, the researchers examined the participants' apparent emotional state by asking them to recall two types of experiences: a frustrating one and an enjoyable one that they had during the past seven days. The interviews were videotaped, later transcribed and the response to each prompt was saved in a separate document. The interviews were conducted at the following intervals: at the baseline, after six weeks, and then two months and six months, post-intervention. Participants talked about typical activities with a focus on their social life and interactions.

Since extracting features using LIWC requires at least some words to count in order to correlate with the built-in emotion categories, we created one dataset containing only documents with at least 130 words to be included in our models, resulting in a document collection of 366 positive and negative interviews. Documents contained an average word count of 303 words, with the largest containing 1732 words and the smallest, 131. We also created a dataset of 448 documents containing documents with an average word count of 258 words, with the largest containing 1732 and the smallest, 2. We used this to see how well small documents were classified by the LDA and LIWC models. To elicit responses containing enjoyable or frustrating content, the interviewer used the following prompt: talk about an enjoyable/frustrating experience that happened in the last week.

3.3.2 Model Design

The overall approach to the model design consists of two processing steps: (1) extract the features from each document in the set, and (2) use these features to predict whether the interview described a frustrating (negative) or enjoyable (positive) experience. The difference between our model and LIWC is the feature extractor used in step (1): LDA topic proportions vs. LIWC word count. The design of the LDA feature extractor is shown in Figure 1 and that of LIWC in Figure 2. Prior to extracting features from the document set, the collection is split using a 90/10 proportion into a training and test set, shown in steps 1 and 2 in both figures. This is to create "set-aside" test documents, which can be used to evaluate how well the model predicts whether an interview is enjoyable or frustrating for a document that has not been used for training. The training set is used to build the generative topic model which is then used to infer the topic proportions (i.e., features) of both the training set as well as the set-aside test set. Once the training and test sets have been created, feature extraction follows two distinct processes for LDA and LIWC.

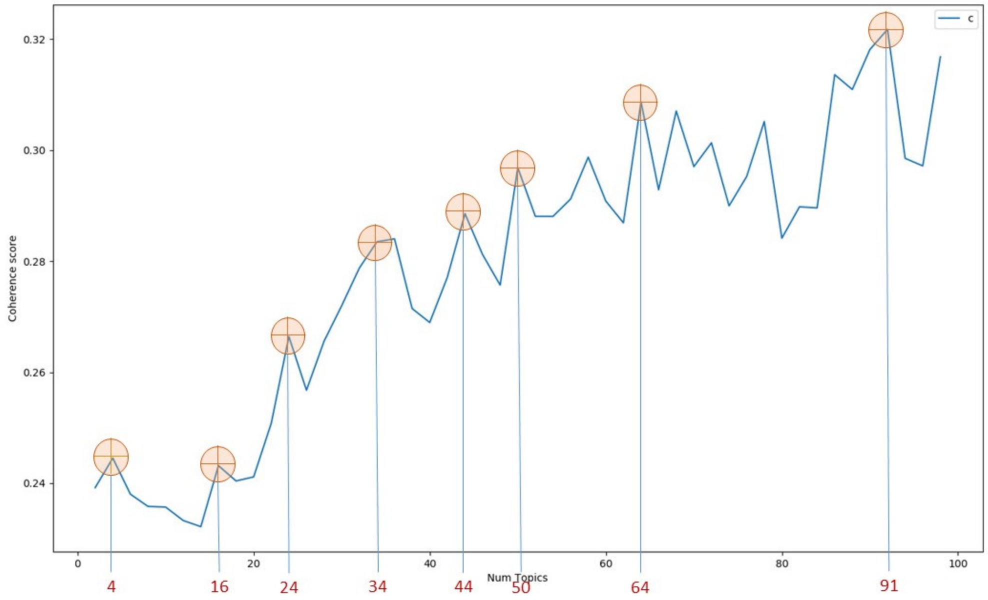


Figure 3.1: Coherence score by number of topics. The LDA model was trained repeatedly using the training set, starting with 2 topics. At the end of each iteration, the coherence score was calculated and the number of topics was increased by 2 until 100 topics was reached. Eight local maxima are identified by cross-hairs on the graph.

3.3.3 LDA Training and Feature Extraction

Once we have split our document collection, we can use the training set to generate the topic model and infer its thematic structure, i.e., topic proportions. We use the Gensim [Řehůřek and Sojka, 2010] implementation of LDA, a robust, stable version that is widely used in academic research for topic modeling and natural language analysis. While it is possible to adjust many of the implementation's parameters (e.g., the Dirichlet priors for the per document distributions and for the per topic word distributions), we accepted the default values. As mentioned earlier when we introduced LDA, the generative model assumes a number of topics over which an initial distribution of documents (i.e., $P(\Theta_d)$) is estimated. We now describe how we selected the number of topics.

Recall that a topic model tries to discover a thematic structure in a document collection; it is trying to find structure in otherwise unstructured text. One of the characteristics of this type of machine learning method is that it does not guarantee that the topics will be interpretable by humans. Thus a measure is needed to automatically evaluate the topic quality of the topics generated by the LDA model. We use the topic coherence pipeline available in Gensim which is an implementation of the method described by Röder et al. [2015]. In the context of topic modeling, a coherent model is one in which words are treated as facts; coherence can then be evaluated on the basis of how well the words in a topic "support" one another, as when we speak of a coherent set of facts. In the topic model, words support one another based on their probability of co-occurring together. The coherence measure produced by the framework described by Röder et al. [2015] is a real number representing an aggregation of probability estimates; this number can be used to compare the topic quality of different topic models.

The researchers report that the model has been extensively compared with human gold-standard coherence measures using Wikipedia as a reference corpus and has performed quite well. Figure 3.1 shows a plot in which the LDA model was run with an increasing number of topics in steps of 2, from 2 to 100, against which the coherence score was calculated. We can identify eight local maximum values at 4, 16, 24, 34, 44, 50, 64, and 91 topics respectively. We hypothesize that the interview process, during which a participant was asked to recall a frustrating and enjoyable activity, tends to generate a large set of words with different co-occurrences across participant interviews. However, there are a set of topics which distinguish between frustrating and enjoyable content, allowing the model to use these topics to predict emotion valence. We will describe how these eight topic-number values were used in the model evaluation in a subsequent section.

As shown in step 2 of Figure 3.2, once we have split the interview transcriptions into training and test document sets, we set aside the test set and proceed

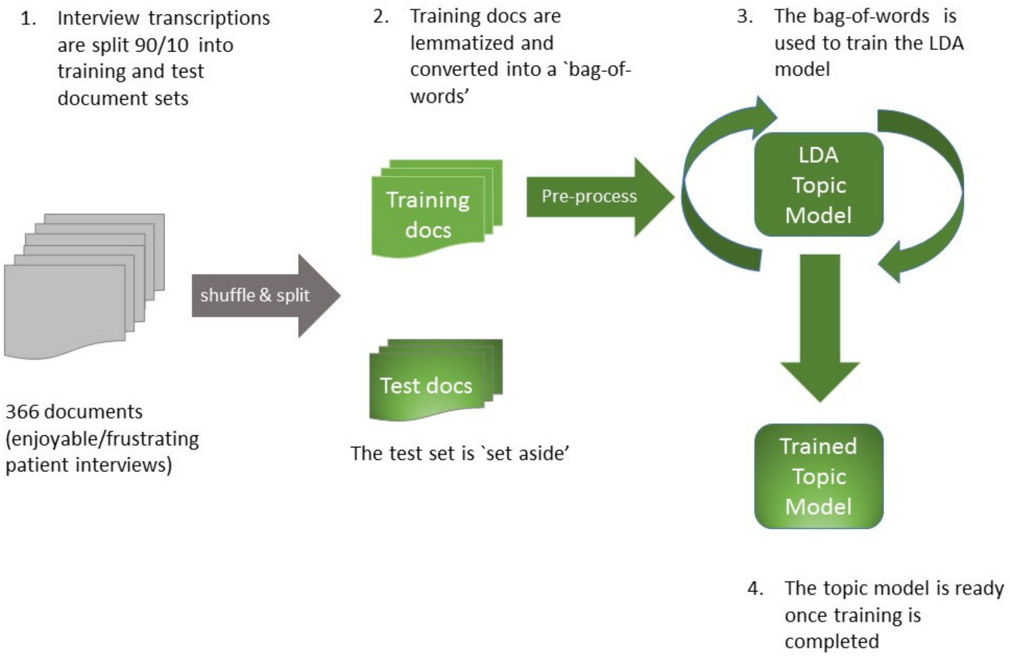

1. Interview transcriptions are split 90/10 into training and test document sets
2. Training docs are lemmatized and converted into a `bag-of-words'
3. The bag-of-words is used to train the LDA model
Training docs
Pre-process
LDA Topic Model
shuffle & split
Test docs
366 documents (enjoyable/frustrating patient interviews)
The test set is `set aside'
Trained Topic Model
4. The topic model is ready once training is completed

Figure 3.2: LDA feature extractor.

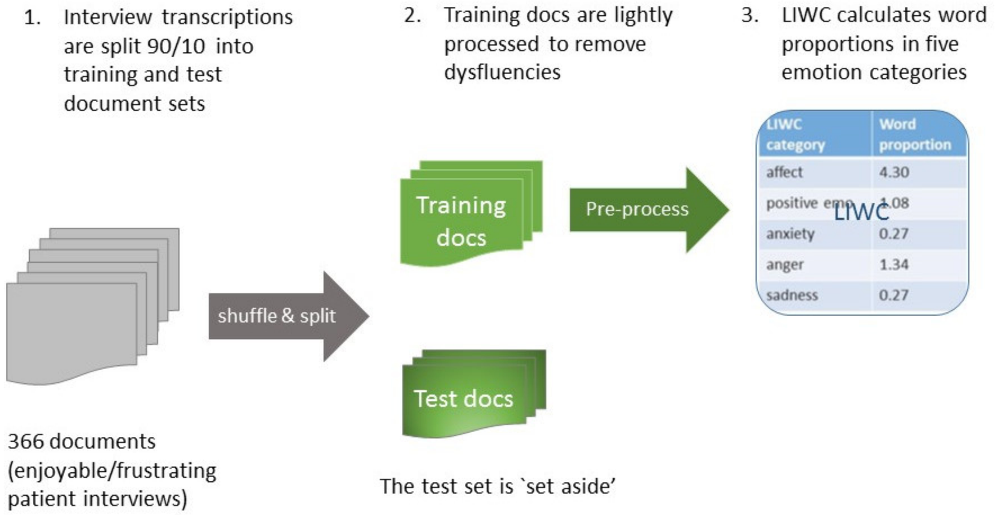

1. Interview transcriptions are split 90/10 into training and test document sets
2. Training docs are lightly processed to remove dysfluencies
3. LIWC calculates word proportions in five emotion categories
LIWC category
Word proportion
affect
4.30
positive emo
1.08
anxiety
0.27
anger
1.34
sadness
0.27
LIWC
Training docs
Pre-process
shuffle & split
Test docs
366 documents (enjoyable/frustrating patient interviews)
The test set is `set aside'

Figure 3.3: LIWC feature extractor.

to pre-process the training set. The purpose of pre-processing is to transform the original text into a more efficient set of words, removing information that does not help the LDA model infer its thematic structure. Pre-processing lemmatizes words (i.e., words in the third person are changed to the first person; verbs are changed to the present tense) and words are stemmed (i.e., reduced to their root). Common "stop" words (e.g., the, is, at) and disfluencies (e.g., um) are removed. Document text is split into sentences and then into words and word frequencies are computed. It should be noted that during the pre-processing, the ordinal nature of the document structure is broken and it becomes a bag of words. The LDA model does not use grammatical structure to infer thematic structure.

Training is completed once the LDA model has estimated the hidden variables Z (per-word topic assignment) and Θ_d (per-document topic proportion), which the LDA model in Gensim does automatically on our behalf. At this point we have a trained topic model to which we can supply unseen documents and obtain topic proportions; we can also extract the topic proportions already assigned to the documents it used for training. In either case, the topic model produces a set of feature vectors, one for every document the size of each being the number of topics. However, we do not yet have an association between the topic proportions and the classification of a document as "frustrating" or "enjoyable". In a subsequent section, we will describe how we can use a machine-learning tool known as *classifier* to make this association.

3.3.4 LIWC Feature Extraction

Takahashi et al. [2010] used five LIWC dictionaries (categories) to measure participants' verbal expression of positive and negative emotion. They are: positive emotion, anxiety or fear, anger, sadness or depression, and achievement. The researchers used the 2010 version of LIWC to extract word counts in these categories from participant interviews. An analysis of variance (ANOVA) was used to show a statistically significant effect that participants used more words categorized by LIWC as expressing positive emotion when talking about enjoyable activities rather than frustrating activities, and used fewer words expressing negative emotion. Alternatively, participants used more negative words when asked to recall a frustrating activity. Thus we chose these five categories to be used as features, hypothesizing that they could be associated with the classification of an interview as being frustrating or enjoyable. As shown in step 2 of Figure 2, the participant interviews were lightly processed to remove disfluencies and then input to the 2015 version of the LIWC software. The resulting output is a set of feature vectors for every document, each vector of size five and where each feature represents the word proportion of the corresponding LIWC emotion category. The feature vectors generated by LDA and LIWC were then used by a classifier to learn the association between the features and the type of interview.

3.3.5 Using Features to Predict Emotion Valence

In machine learning, a classifier is a software tool used to predict classes of items rather than values; the latter is performed using regression techniques. In our model we use a logistic regression (LR) classifier to predict a set of two possible interview classes, $interview = frustrating, enjoyable$. We use a stable, widely-used implementation of an LR classifier from Scikit-learn, a free software machine learning library [Pedregosa et al., 2011]. Logistic regression, developed by statistician Cox [1958], computes the probability of output in terms of input and this can be used to construct a classifier by choosing a cut-off probability value (i.e., 50%) and classifying input values greater than the cut-off as one class and below the cut-off as the other. The classifier is trained and used to predict the interview classes in exactly the same way for both the topic features and LIWC features (see Figure 3.4); the only difference is the feature set used, and the following discussion holds for both sets.

Training the logistic regression classifier consists of finding the parameters θ of the model such that it sets high probabilities for enjoyable content and low probabilities frustrating content. This is achieved by minimizing the cost function, $c(\theta)$, where the probability estimate is $\hat{p}$ and the training label is y:

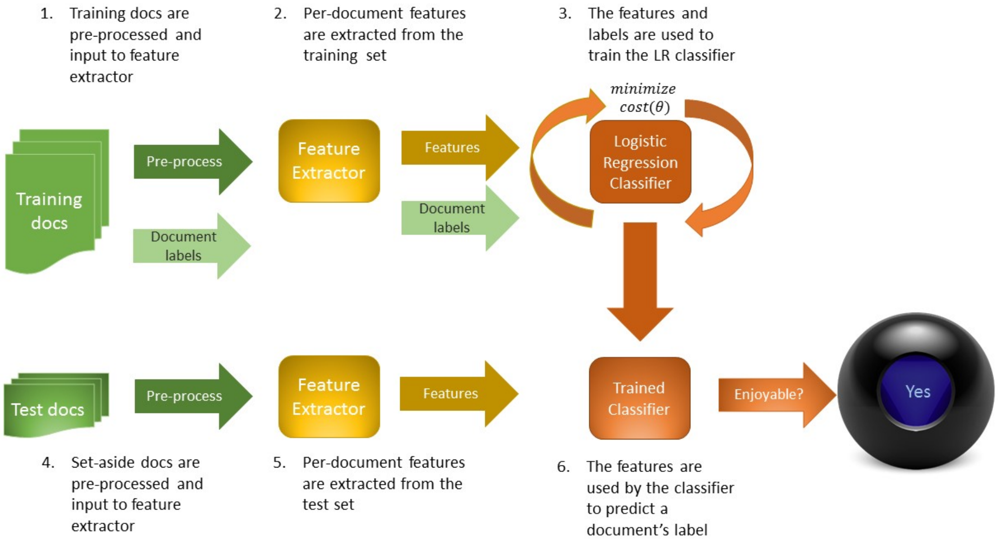

1. Training docs are pre-processed and input to feature extractor
2. Per-document features are extracted from the training set
3. The features and labels are used to train the LR classifier
minimize cost(θ)
Training docs
Pre-process
Document labels
Feature Extractor
Features
Document labels
Logistic Regression Classifier
Test docs
Pre-process
Feature Extractor
Features
Trained Classifier
Enjoyable?
Yes
4. Set-aside docs are pre-processed and input to feature extractor
5. Per-document features are extracted from the test set
6. The features are used by the classifier to predict a document's label

Figure 3.4: Logistic regression classifier. 1,2) Features are extracted from the training set using either LDA or LIWC. (3) Training proceeds until the model minimizes a cost function, $c(\theta)$, which penalizes misclassification. (4,5,6) Features are extracted from the set-aside documents and presented to the trained classifier for predicting the interview type (yes = enjoyable, no = frustrating).

$$c(\theta) = \begin{cases} -log(p) & \text{if } y = 1 \\ \\ -log(1-p) & \text{if } y = 0 \end{cases} \tag{3.1}$$

Consistent with the goal of the classifier, $log(x)$ grows larger when x approaches 0, and therefore, the cost will be large if the classifier estimates a probability close to 0 for an enjoyable interview; likewise, it will also be large if it estimates a probability close to 1 for a frustrating interview. Alternatively, $-log(x)$ is close to 0 when x approaches 1 and the cost will be close to 0 when the estimated probability is close to 0 for frustrating interview and close to 1 for an enjoyable interview. To compute the value of θ that minimizes the cost function, the Scikit LR classifier implementation uses an optimization method known as stochastic gradient descent (a good discussion can be found in [Géron, 2017]). Once classifier training was completed, we evaluated the LDA and LIWC models' performance using materials from [Tickle-Degnen et al., 2010].

3.4 Results and Evaluation

Table 3.1: Experiment 1: Model evaluation using 10-fold cross-validation for 332 training documents with average word count = 303; max = 1732; min = 131

LDA evaluation

Features	Precision	Recall	F_1
4	0.56	0.56	0.58
16	0.54	0.55	0.54
24	0.62	0.62	0.62
34	0.62	0.63	0.62
44	0.59	0.59	0.59
50	0.57	0.57	0.57
64	0.57	0.58	0.57
91	0.63	0.63	0.61

LIWC evaluation

Features	Precision	Recall	F_1
5	0.74	0.74	0.74

Table 3.2: Experiment 1: Model testing using 34 documents with average word count = 303; max = 1732; min = 131.

LDA evaluation

Features	Accuracy
4	0.71
16	0.65
24	0.59
34	0.68
44	0.65
50	0.53
64	0.56
91	0.74

LIWC evaluation

Features	Accuracy
5	0.76

3.4.1 Experiment 1: Predicting Interview Class Using Larger Word Counts

For this evaluation, we used the document collection, from the Self-management Rehabilitation and Health-Related Quality of Life in Parkinson's disease database where the average word count = 303 to train and test the model; there are 332 and 34 documents in the training and test sets respectively. The LDA feature extractor (see Figure 3.2) as trained eight times by setting the LDA model's parameter for the number of topics according to the eight values identified by the coherence model as local maxima (see Figure 3.1. Each training session i, where $1 \leq i \leq 8$, and $n_i = \{4, 16, 24, 34, 44, 50, 64, 91\}$ topics generates a feature vector of size n_i for each document in the training set. Each feature vector is associated with a document's target label (i.e.,, $enjoyable = 1, frustrating = 0$) and the $(feature, target)$ pair is used to train the logistic regression (LR) classifier using a method known as *K-fold cross validation.* The results for each training session are shown in Table 3.1. In K-fold cross validation, the training set is split into K distinct subsets called folds. We set $K = 10$; this is typical for the size of our training set, which is considered small compared to typical machine learning problems that can have several thousand training instances. This process trains and evaluates the LR classifier ten times choosing a different fold for testing every time and training on the remaining nine folds.

We include more robust metrics than accuracy to evaluate the model performance: precision, recall, and F1. Precision gives a measure of the accuracy of positive predictions. It is computed as follows, where TP is the number of true positives and FP is the number of false positives. Thus a model with a low precision will tend to signal a high number of "false alarms". It is often used with another measure called *recall*, also known as *sensitivity* or the *true positive rate*, the proportion of positive instances correctly identified by the model. It is computed as follows, where FN is the number of false negative instances.

$$\text{precision} = \frac{TP}{TP + FP} \qquad\qquad \text{recall} = \frac{TP}{TP + FN}$$

For example, referring to Table 3.1, when using four features and when the model predicts that the interview is enjoyable, it is correct only 56% of the time; when an interview is enjoyable, it predicts so 56% of the time. It is common when evaluating classifier to combine precision and recall into a single statistic, called the F_1 score. This score is the *harmonic mean* of the precision and recall, which unlike the arithmetic mean, balances both; you cannot get a good F_1 if either are low. F_1 gives its best score at 1, when precision and recall are perfect. Thus, the harmonic mean will generate high F_1 values when both the precision and recall are high. We can see, for example, that the F_1 score of 61 is highest when features = 24, 34.

As discussed, we have trained eight LDA feature extractors, corresponding to the number of topics we presented as a parameter to the model and that we have set-aside 10% of our documents (i.e., 34) which have never been used to train either the LDA feature extractor or the LR classifier. For each of these feature extractors, we present the test documents in order to extract their features and then we present them to our trained classifier which predicts whether the documents are either frustrating or enjoyable interviews (see Figure 3.4). The results are shown in Table 3.2 which gives the classification accuracy for each feature set size used. The accuracy is the mean score for across the 34 test documents.

The five emotion categories used by Takahashi and Tickle-Degnen [Takahashi et al., 2010] were used to extract features from the 332 training documents as shown in Figure 3.3. These feature vectors were paired with their corresponding document target labels and we followed the same 10-fold cross validation procedure described in the previous section to train the LR classifier (refer to Figure 3.4). The precision, recall, and F_1 evaluation metrics are shown in Table 3.1. We then used the LIWC categories to extract the features from the set-aside test documents and presented them to the trained LR classifier in order to predict each document's interview category. The results are shown in Table 3.2 which gives the mean classification accuracy across the 34 test documents.

Table 3.3: Model testing using 14 documents with average word count = 13; max = 22, min = 2.

LDA evaluation

Features	Accuracy	Precision	Recall	F_1
4	0.79	0.75	0.86	0.80

LIWC evaluation

Features	Accuracy	Precision	Recall	F_1
5	0.64	1.00	0.29	0.44

3.4.2 Experiment 2: Predicting Interview Class Using Smaller Word Counts

In this experiment, we investigated how well an LDA model trained on a collection of 404 documents whose word count ranged from 2 to 1732, with an average of size of 258 could accurately predict the interview class using test documents representing short bursts of dialog. For the test set, we used transcripts that were edited to obtain the first 20 seconds of conversation. These documents have an average word count of 13 words, and ranging from 22 to 2 words; this is the typical word count found in the average sentence. The LDA feature extractor was trained using number of topics equal to 4.

As can be seen in Table 3.3, the LDA model F_1 score of 0.80 using the test set is considerably better than the LIWC model score of 0.44. Presumably this is because it has extracted the topics from the context of all the documents in the training set and thus is able to use this information to situate the unseen document in these topics. In contrast, LIWC only uses the words in the test document presented to it which may contain insufficient content to accurately predict the emotion content. We do not report statistics for the training process since the training data is similar to what was used for Experiment 1 and the purpose was to evaluate short test documents.

3.5 Discussion

The findings suggest that LDA can be used to discover a set of topics whose proportions for any given document in the collection can be used to parsimoniously represent the positive or negative emotion content of that document. It appears that the number of topics used to extract the features does not greatly affect the performance of the classifier. The most compact feature set using four topics, had an F_1 score of 0.58 and resulted in a test set accuracy of 71% whereas the largest feature set of 91 topics had an F_1 score of 0.61 and a test set accuracy of 74%. In comparison, the LIWC model used five categories of words which have been previously shown to correlate with interview category [Takahashi et al., 2010]. Words belonging to each category were tallied and used to calculate their category's proportions. This approach produced an F_1 score of 0.74 and a test set accuracy of 76%. When faced with a number of choices of increasing complexity, all of which have similar explanatory power in a model, it is reasonable to choose the least complex explanation (i.e., Occam's Razor). The least number of topics that can be used to train the LDA model and generate features that separate the documents into positive and negative emotion categories is four and in the remainder of the discussion, we will assume this version of the LDA feature extractor.

Classification accuracy is only one part of the evaluation; precision and recall, described earlier, are metrics that provide more nuanced on the models' predictive behavior. In Experiment 1, precision and recall are both 0.56 during the cross-validation training. This gives us insight as to how the model will perform across a variety of test sets. These metrics suggest that LDA does better than chance predicting the interview category in two situations: (1) when it makes a prediction, it is correct 56% of the time (precision) and (2) given the actual target, the model makes the same prediction 56% as well (recall). The LIWC model, with an F_1

score of 0.74, performs better in both cases, 74% of the time. We note that for both models, precision = recall and thus neither is biased toward giving more 'false positives' (precision) or 'false negatives' (recall). These statistics suggest that the LIWC model would more accurately classify new interviews whose average word count is 303 words (approximately 22 sentences per document on average). The performance differential may be due to the five dictionaries selected to be the source of the features used in the LIWC model.

On the other hand, in situations where the model is likely to encounter one or two sentences, Experiment 2 (Figure 3.3), which produced an F_1 of 0.80, suggests the LDA to be the better choice. We might, for example, create a communication assistive tool to infer the emotion content of each interaction between a person with PD and a caregiver, which most likely consists of short bursts of dialog. In this experiment, the precision and recall metrics have different values in both LDA and LIWC. In LDA precision = 0.75 and recall = 0.86 suggesting that the model is biased away slightly from making false negative predictions and more towards false positives. In an interaction with a PD person and a caregiver, the LDA model will evaluate that turn in the conversation to contain more positive emotion (more enjoyable) when it may in fact contain more negative emotion (more frustrating), slightly more frequently (11%) than making a negative prediction when the turn is positive. However, the LIWC precision of 100% suggest that it will almost never make a false positive prediction, but when it does makes an incorrect prediction, which it did in this experiment, 1.00 - 0.64 or 36% of the time, it is likely to be a false negative. A higher level of false negative predictions is an example of the desynchronization of mental state and facial expression which Tickle-Degnen et al. [1994] report "Can have a profound effect on communication ability and quality of life".

Another characteristic of the LDA model is that it is not sensitive to the language of the text and uses the entire document collection during training to infer the hidden topic structure. Thus it can learn thematic structure from documents in any language and use what it has learned to place documents it has not seen in the topic structure, even short, one or two sentence documents. LIWC, however, would have to be modified to incorporate a new language and its dictionaries would have to be updated to ensure the new language's words were placed in the proper emotion categories. Thus we suggest superior performance of LDA in spoken dialog of persons with PD. Persons with PD have difficulty with enunciation and voice volume, therefore automated speech recognition technology (ASR) at its current level, 15% Word Error Rate, is likely to produce inaccurate transcripts. This will make it difficult for LIWC to recognize words and process word counts. Since LDA is not sensitive to the word orthography, it should still be able to use the imperfect transcriptions to extract the topic features; this theory remains to be tested. At present, we used the model to classify emotion valence categorically as enjoyable (positive) or frustrating (negative); however it is also possible to use the model to predict other discrete emotions including the level of arousal. In a future version of the model, predicting both valence and arousal can be used to observe and inform the emotion trajectory of a conversation as it unfolds between any two participants, for example, in therapy or counseling sessions.

3.6 Limitations

The results of this study suggest that topic modeling could extract features associated with emotion valence using verbal transcriptions of interviews in which participants were specifically asked to recall an enjoyable and a frustrating experience. We shall point out a few limitations of this approach. People living with PD might have talked about frustrating things when describing enjoyable experiences, this sometimes happens in the context of chronic illness, especially with people with depression. However, our participants were screened for depression and were found to not be clinically depressed.

Also, human emotional state can change in far more complex ways and in more subtle gradations than the positive/negative emotional categories we have explored in this study. In addition, further research in how humans combine input from several modalities such as visual, auditory, and tactile to generate understanding of complex mental states such as embarrassment, thinking or depression is needed to inform a more naturalistic and incremental model. Furthermore, the topic model infers the hidden thematic structure and the topics are not easily interpretable. The topics cannot really tell us much in a way that makes sense to a human why a certain text has an enjoyable or frustrating emotional content. Finally, our training and testing dataset was comparatively small. At most, we had 448 transcripts in the document collection, a limited amount of data compared to typical machine learning endeavors which may have thousands of training examples available. This means that in order to generalize the test results to a domain beyond that described in Tickle-Degnen et al. [2010], additional training documents will have to be used.

The initial purpose of this research was to investigate whether interview data from persons with PD could be used to train a model to predict whether a new utterance described something enjoyable or frustrating. We view this as a first stage in building an assistive tool which a caregiver could use to infer the emotional state of a person with PD. In order for the model to be useful in a clinical setting as a communication assistive tool, it will have to be developed further to generate the more incremental and subtle gradations of human emotional states. In addition, clinical trials would need to be conducted to assess its usefulness.

3.7 Summary

We investigated an automated method for inferring the emotional state of a person with Parkinson's disease using a machine learning approach. Our results show that the LDA model performs better with shorter text which makes it more suitable for evaluating emotional content of short dialog turns. For longer documents, LIWC performs better; however, it has the shortcoming of assuming a constant, non-evolving language and is dependent on manually selecting the dictionaries to be used as features in the problem domain. It is also non-generalizable to other languages for which dictionaries have not yet been created.

The LDA model is a first step towards creating an assistive tool which the caregiver can use to infer the emotional state of a person living with PD. Given that many people with PD live in the community, the caregiver is likely to be a member of the family who is not necessarily trained or accustomed to the symptoms of the disease and may make incorrect inferences about the person's true emotional state. Thus an assistive tool equipped with the ability to accurately and immediately provide feedback on the emotion content of a conversation is not only beneficial for improving the social interaction with the PD patient, it can improve the quality of life in the home, including that of the family members. This capability can also be to assist the rehabilitation therapist in, for example, during client evaluation, helping preserve the client's dignity in situations where the client's claims to be happy is belied by her affectless face. This technology is not restricted to the domain of people living with Parkinson's disease; it should be able to generalize and serve as an intelligent agent useful for monitoring the emotional content of the interaction between any two parties, providing real-time feedback on the emotional content as the interaction unfolds

Chapter 4

Detecting Emotion Polarity from Continuous Speech in a Robot

In this chapter we discuss how we used the experience gained from detecting sentiment from transcriptions of the interviews conducted with the Parkinson's disease persons to build a robot that can detect emotion in continuous speech and express it though facial expressions. We did this in two parts: (1) infer sentiment from transcription of the PD person's utterance, and (2) extend this to use utterances generated from continuous speech, in real-time.

In the first step, we defined a three-state emotion detector that can predict positive, negative, and neutral emotion from utterances that have been transcribed from the PD interviews and used it to drive three facial expressions on Robo Motio's Reddy robot: smile, frown, and neutral. As we did not have emotion labels for the individual utterances of the interview transcriptions, we built a Web-based tool called the Emotional Inference Topic Model (EMIT) which presents lines of text one at a time to a human evaluator. The trained model was integrated into a robotic cognitive architecture to perform real-time, continuous speech detection of positive, negative, or neutral emotional valence that is expressed through the facial features of a humanoid robot. We tested the model using the ground truth labels obtained from human evaluators. This formed the basis for developing a more robust model with finer prediction resolution. In order to train the model, we needed to collect written text which is labeled incrementally (e.g., sentence by sentence).

In the second step, we incorporated a Large Vocabulary Automatic Speech Recognizer (LVASR) into the robotic cognitive architecture in such a way that it can transcribe ordinary conversational speech in real time and present the transcribed utterances to the emotion detector. We used an already-trained acoustic and language model that was developed using the Kaldi toolkit. This model performs near state-of-the-art with a word error rate ("WER") of approximately 13%. As originally developed, input to the model is a .wav file; we adapted it to the on-line ASR pipeline that DIARC uses. Once integrated, we evaluated the entire emotion pipeline by using a human speaking to the robot and judging how well the robot's facial expressions match was it is being said. For ethical reasons (i.e., we would have to recruit heavily from a vulnerable population), we did not use people with PD for these initial stages. However, a clinical trial is advisable and desirable in the future. For this reason, an actor read from the actual transcripts of the PD person's interviews and this was be recorded. The recordings were evaluated by Amazon Mechanical Turk workers.

4.1 Introduction

Our prior research, described in Chapter 3 suggests that sentiment lexicons can be ineffective when trying to track the emotional content of a conversation as it unfolds (also see [Valenti et al., 2019c]). Approaches such as LIWC are not optimal when using as input short pieces of text with few words, which is often the case with speech utterances. We described a novel method to automatically track the progression of the emotional content in a conversation. The method tries to fit the utterance into the thematic structure of a set of documents on which the model was previously trained. One might think of these documents as representing the conversational domain in which the intelligent agent can be expected to operate. We used the Latent Dirichlet Allocation (LDA) generative model as the basis for an unsupervised learning model, which we trained to extract topic proportions from a collection of written text documents. When an unseen sentence was presented to the model, it found its topic proportions and used them as a set of features. We then used a Multi-Layer Perceptron (MLP) classifier to associate these features with training data labeled according to emotion valence (positive or negative) and arousal (high or low). Zhang et al. [Zhang et al., 2013] used a neural network to detect affect from facial expression and a Latent Semantic Analysis model [Deerwester et al., 1988], a non-generative predecessor of LDA, to detect topics embedded within the human-robot conversation; however, the detected topics were not used to inform affect detection.

This chapter proceeds as follows. Section 4.2.1 describes how the model was trained for detecting emotion using sentences as input and summarizes how the model forms an emotion pipeline in the robotic cognitive architecture. We also describe how the embedded pipeline was tested using transcription of utterances drawn from PD persons. In Section 4.3, we describe how the the model was extended in order to detect emotion from the continuous speech spoken to the robot and transcribed by the ASR. In Section 4.4.3, we explain the human-robot interaction experiment we ran to validate the model. The results of the study suggest that when embedded in the robot, participants recognized the connection between the robot's emotion expressions and the speaker's utterances more when the robot emoted based on the model's predictions than when the robot emoted randomly. Finally, in Sections 4.4.4 and 4.5, we discuss the advantages, disadvantages, and limitations of this approach and the potential for embedding the model in emotion-detecting assistive conversation tools.

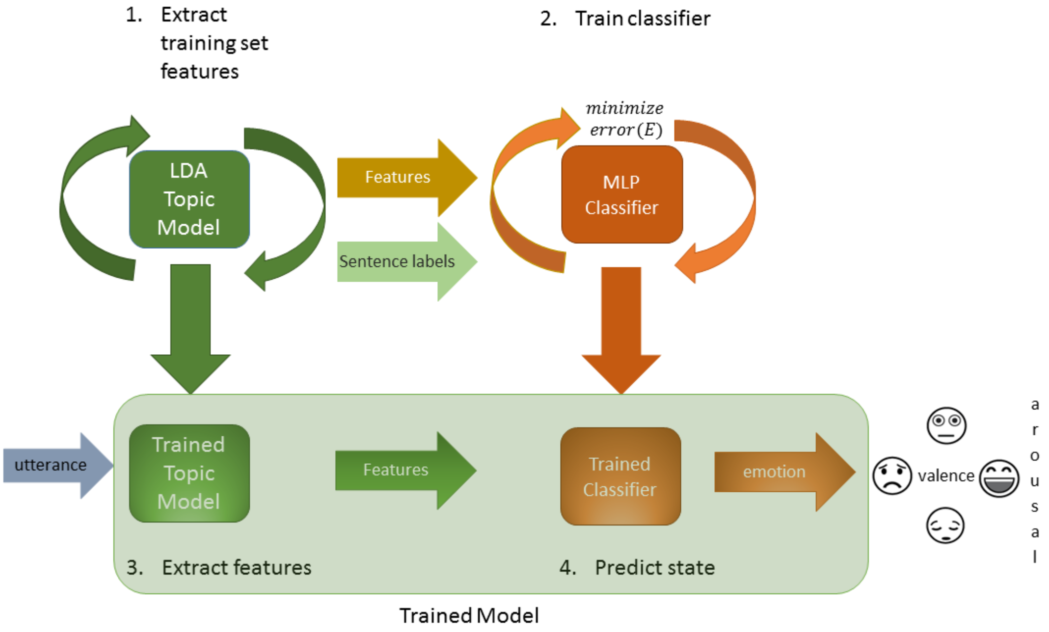


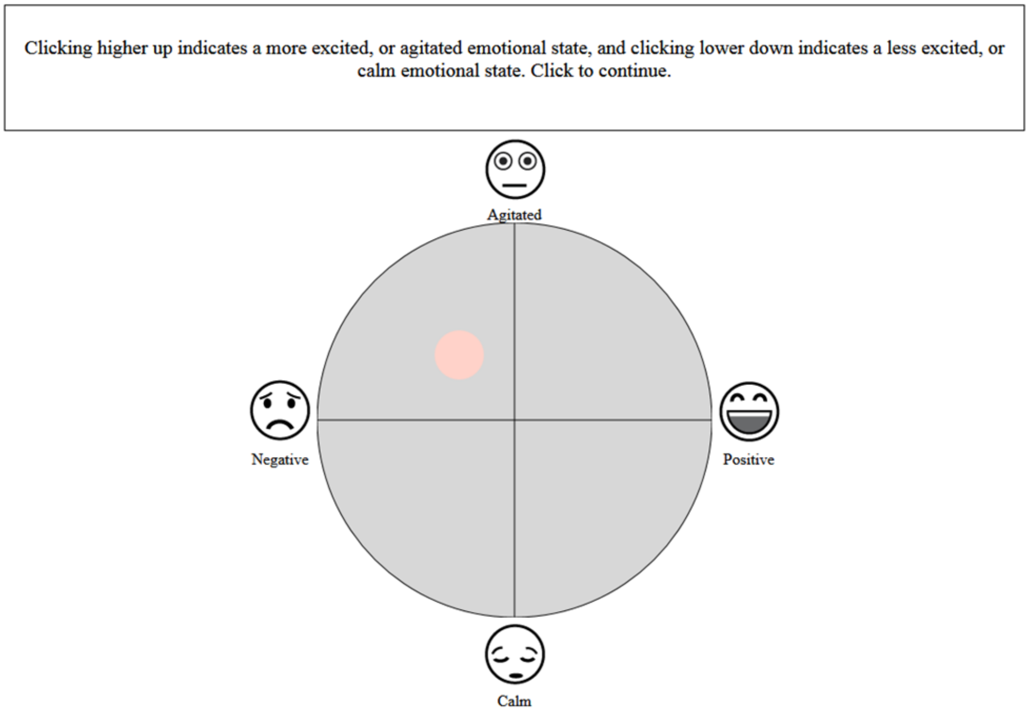
Figure 4.1: Affective prediction model. (1) The LDA model is trained to extract features from the interview documents (2) For each document's sentence, its topic proportions (features) are extracted and, along with its emotion target, is used to train the classifier (3) Features are extracted from the utterance by the trained LDA model, and (4) presented to the trained classifier to predict its emotional state

4.2 Detecting Emotion in Sentences

4.2.1 Methods

Our model was trained on the individual sentences drawn from 448 documents with an average word count of 258 words, the largest containing 1732 and the smallest, 2 as described in Chapter 3. The documents were constructed from selected interview transcripts from 106 male and female participants with PD, living in the community, who participated in a study [Tickle-Degnen et al., 2010] which asked them to recall two types of experiences: a frustrating one and an enjoyable one that they had during the past week. Thus, the robot could be expected to accurately predict emotion from utterances spoken in a similar contextual domain.

We used the interview documents to collect ratings of the emotional content (i.e., valence and arousal) for each sentence of the document; these values were used as the training targets for the model. We collected this data using Amazon



Clicking higher up indicates a more excited, or agitated emotional state, and clicking lower down indicates a less excited, or calm emotional state. Click to continue.

Figure 4.2: Human-labeling of text using EMotion Inference Tool: EMIT. The mouse is used to position the colored cursor anywhere in the field circumscribed by the emotional circle. At a mouse-click, the tool records the valence & arousal coordinates of the cursor and the elapsed time.

Mechanical Turk (AMT). AMT workers used a Web-based application we created to indicate their perceived emotion contained in text content (see Figure 4.2). To ensure high-quality of the training data, we only used those sentences for which at least 80% of the raters agreed on the label for valence (positive or negative) and arousal (high, low). For valence this represented 1,058 sentences; for arousal 615 sentences. This shows, as expected, that humans had more difficulty inferring arousal than valence in this dataset.

The model design consists of two processing steps: (i) extract the topic proportions from each document in the set (items 1 and 3 in Figure 4.1), and (ii) use these features to predict the emotion valence and arousal of individual sentences as yet unseen by the model (items 2 and 4 in Figure 4.1). Training of the LDA model and the classifier (items 1 and 2 in Figure 4.1) was done outside of the robotic architecture and the results saved to files. These were subsequently used to initialize the emotion pipeline. In principle, training could also take place in the robotic architecture.

We trained the LDA topic model to generate vectors with 34 features. We used a Multi-layer Perceptron (MLP) with one hidden layer and 34 artificial neurons to associate the features produced by the trained LDA model with the training targets collected from the human workers. Further detail on the model design and parameter selection is given in [Valenti et al., 2019a]. MLPs can be configured as multi-label classifiers which would allow us to configure the model to predict the valence and arousal values either separately or jointly from a given feature vector. As previously mentioned, predicting both valence and arousal simultaneously proved problematic because there were relatively few training examples in which there was high agreement for both valence and arousal in the same sentence. For the initial phase of the investigation reported in this paper, we chose to predict valence only as the state where there was high agreement among the human raters and had many more training examples.

Even though we collected real-number values for valence and arousal from the AMT workers, we used the MLP as a binary classifier rather than as a regressor.

As a result, we took the mean of the raters' valence (x-values) and converted all positive real-number values to '1' and negative values to '0'; these were used as training targets for the classifier. If the binary value of the prediction is 0, it was interpreted as negative valence and if was 1, it was interpreted as positive valence.

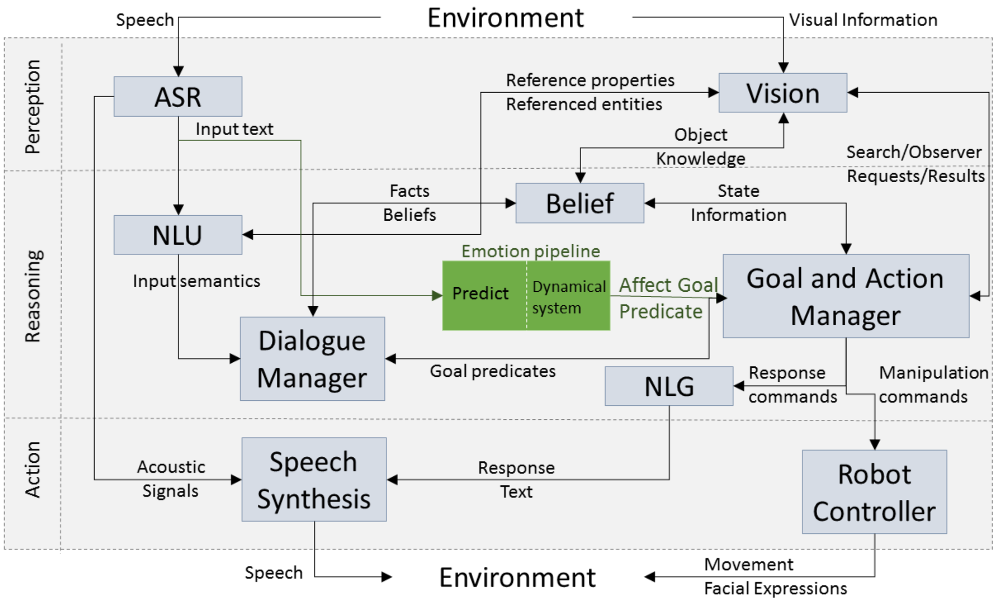


Figure 4.3: The Emotion Pipeline consists of the prediction and dynamical systems components. It receives utterances from the Automated Speech Recognizer (ASR) and sends its predicted affect to the Goal and Action Manager.

4.2.2 Architecture

Our supplemental emotion pipeline consists of two components, the *Predict* component and the *Dynamical System* component, along with their incoming and outgoing connections (refer to Figure 4.3). The Predict component receives a text utterance, extracts its topic proportions and gives these features to the classifier which makes a prediction as to the current emotional state (this corresponds to items 3 and 4 in Figure 4.1). The prediction is then passed to the Dynamical System component which, at present, simply maps the prediction, $\{0, 1\}$ onto a goal predicate string as $\{0 : frown, 1 : smile\}$. The Dynamical system component, described in Chapter 5, provides a means to smooth prediction errors; it sends the goal predicates representing desired affective states to the Goal Manager.

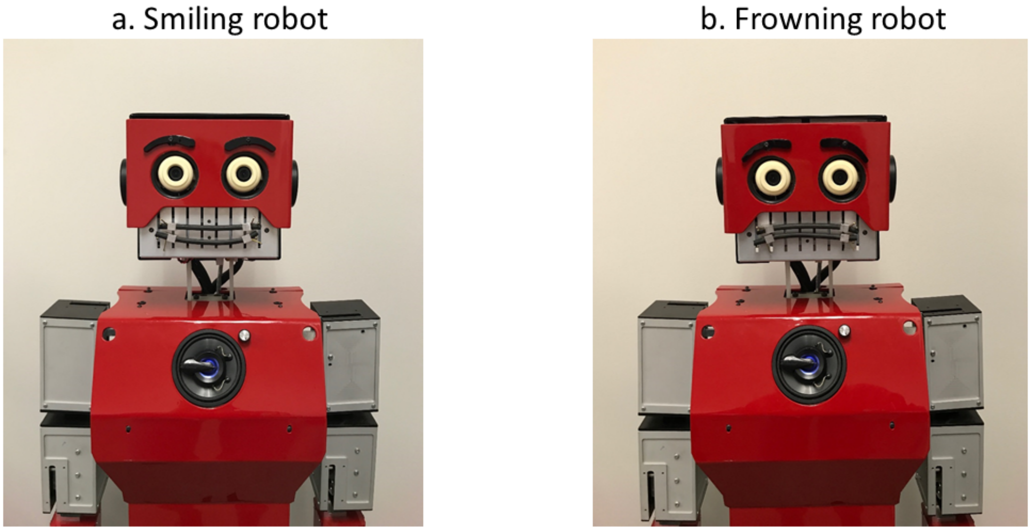


Figure 4.4: Robo Motio's Reddy robot. a) Smiling b) Frowning

The Goal Manager maintains the systems' beliefs about goals within the world and uses those beliefs to decide upon the proper series of actions to take to reach those goals. Actions the Goal Manger can execute are represented in a script based format. When an affective goal predicate is submitted to the Goal Manger, it uses an affective control script to select the appropriate *primitive action* which sends messages to the robot controller component to move the robot's actuators to perform pre-specified behaviors.

Here, we are using Robo Motio's Reddy robot (see Figure 4.4). We defined action primitives for "smile" and "frown", which correspond to the robot's facial motors moving to produce a smile or frown, respectively. The action script checks to see if the goal predicate created a "smile" goal or a "frown" goal, and performs the proper action primitive. Through this pipeline, therefore, our robot can change its facial expression to match its belief about the world, as predicted by our model.

Using the cognitive robotic architecture to implement the emotion pipeline gives us flexibility in how it can be used and insulates the pipeline from the implementation details of, for example, the robot's affector motors or the type of speech recognizer used. Furthermore, sending the prediction to the Goal Manager gives the Reasoning system an opportunity to coordinate the robot's facial expression with

Table 4.1: Pipeline demonstration to predict valence using 16 sentences, unseen during model training. Ground truth obtained from humans who rated the content as 1 = postive or 0 = negative emotional valence.

No.	Test Sentence	Ground Truth	Predicted Affect
1	I've fallen and I cant get up.	Neg	frown
2	I am sure that he does not like to shop much	Neg	frown
3	Well, not right now because I cannot do a lot.	Neg	frown
4	Ahh well, we went off to do this and it turn out to be a big ripoff	*Neg*	*smile*
5	Yes, its, uh gotten more, it was, I didn't tell the doctor about it because it came and went.	Neg	frown
6	My frustration is watching my husbands frustration.	Neg	frown
7	Well, I guess a complete lack of mobility	Neg	frown
8	And there isn't much we can do because perception, a persons perception becomes truth to them	Neg	frown
9	Um, I love to read.	Pos	smile
10	The more things that I do on my own instead of having people assist me, I find satisfying.	Pos	smile
11	Well, actually I loved having many children	Pos	smile
12	Im retired for a year now.	Pos	smile
13	I went to a concert yesterday. We took the boat out on the lake out this morning.	Pos	smile
14	But just so when you go in there and you ask people for help some of them are helpful.	*Pos*	*frown*
15	Well, I enjoy driving.	Pos	smile
16	I, what I like doing is physical things like working on my car.	Pos	smile

other activities and to include emotion as additional context in the human-robot interaction.

4.2.3 Demonstration Using Text Input

We selected 16 sentences that had not been used to train the model and presented each as textual input to the pipeline using SimSpeech, a speech simulator component, in place of the ASR. We found that the accuracy of the technology currently used in the ASR component was insufficient to adequately transcribe the sentences used in testing. Swapping ASR components did not require any changes to the pipeline implementation, which is an advantage of using the robotic architecture. The test sentences, their ground truth valence and the predicted robot actions, smile or frown, are shown in Table 4.1. For this test sample, the model correctly predicted the emotional valence in all but two sentences (i.e., numbers 4 and 14), equating to 88% accuracy. Our analysis of why the model makes incorrect predictions is evolving as we gather more experience with different characteristics of our training set. We discuss our thoughts about model performance in the following section.

The results of the demonstration suggest that the emotion pipeline can be used to appropriately generate a smile or frown expression for the robot, and in the Chapter 5, we train a model that automatically detects emotional content with more resolution, moving beyond these coarse levels of valence, negative and positive. Ideally the model could be improved to be able to detect various degrees of positivity or negativity and calm or arousal. That way, the pipeline could be used to inform the emotion dynamics of a conversation as it unfolds between any two participants, such as in therapy. While the relatively high accuracy of 88% seems quite good, further analysis is needed to determine under which circumstances the model makes an incorrect prediction. Preliminary analysis seems to indicate that accuracy is not impacted by sentence word count, but is greatly affected by the number of training examples. The model is also somewhat sensitive the agreement among the human evaluators. We have found that a training set which consisted of sentences in which there was high agreement among the raters was likely to generate a model which was more accurate when given unseen, test sentences.

The method we explored to estimate arousal and valence based on semantics only can, of course, be combined with other components of the robotic architecture, e.g., visual and auditory cues, to get even better estimates, potentially improving the detection of arousal, which we will leave as future work. Additionally, our training and testing dataset was comparatively small. We used 448 transcripts to train the LDA model and 953 sentences to train the classifier, a limited amount of data compared to typical machine learning endeavors which may have thousands of training examples available. We are in the process of collecting additional labeled training examples using AMT for the purpose of developing the model's ability to predict additional emotional states beyond positive and negative. Furthermore, in order to generalize the test results to a domain beyond that described in [Tickle-Degnen et al., 2010], additional training documents from more general domains will have to be used.

4.3 Detection Emotion from Continuous Speech

4.4 Design Challenges

The emotion detection component of our model is sensitive to proper speech to text conversion and end-point detection. As a result, we wanted to ensure that the Automated Speech Recognizer (ASR) used by this component is robust to environmental noise, performs well across a variety of speakers, and could operate "on-line", i.e., transcribe speech in real-time. The ASR is based on the chain model developed for the ASpIRE Challenge [Harper, 2015] and trained on Fisher English that has been augmented with impulse responses and noises to create multi-condition training [Varga, 2017]. The chain model uses Kaldi, a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0 [Povey et al., 2011]. Kaldi has demonstrated low error rates in a variety of challenging acoustic environments using conversational speech [iar, 2015b,a, Harper, 2015].

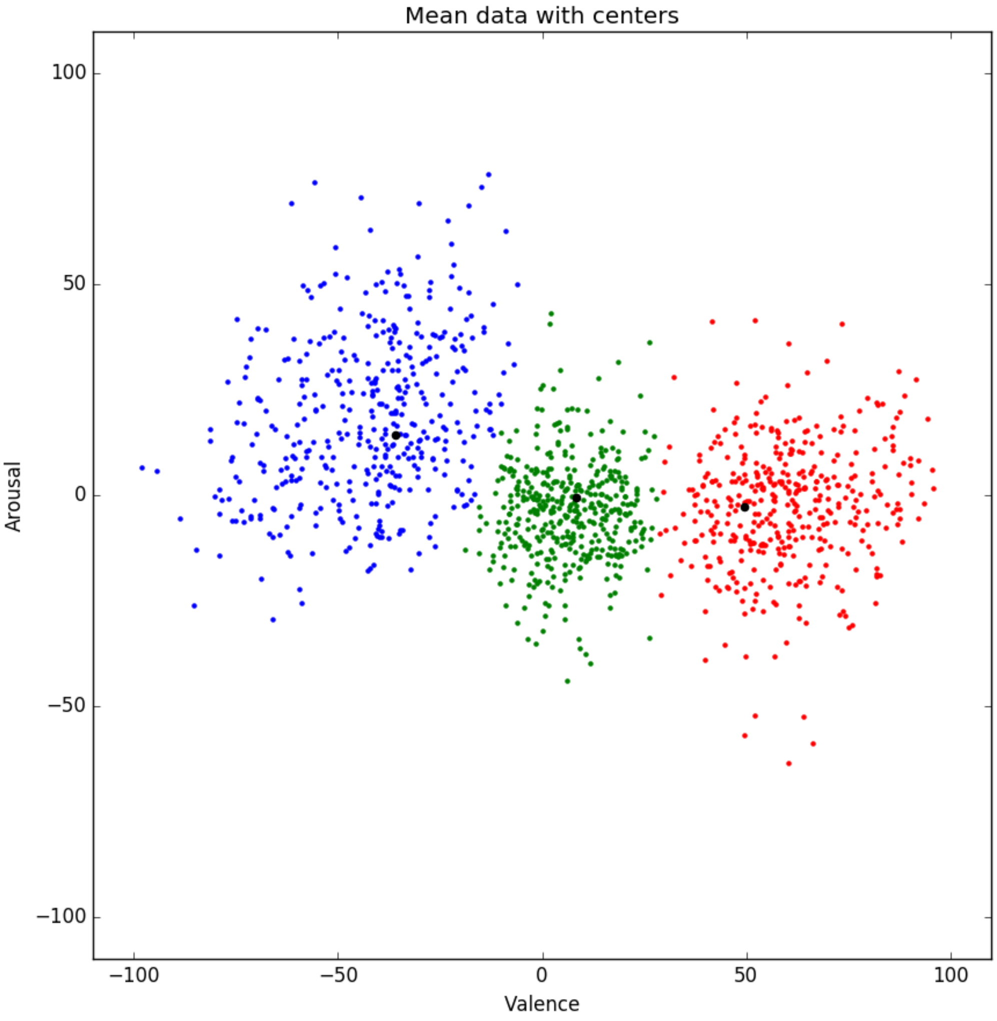


Figure 4.5: k-means clustering identified center-points of the three classes: negative (blue), neutral (green), and positive (red)

The challenge for the ASR is to determine the endpoints of the utterances, in which words are connected together instead of being separated by speech codes such as pauses. Unknown boundary information about words, co-articulation, production of surrounding phonemes, and rate of speech affect performance [Gulzar et al., 2014] and the ASR may generate textual representations that vary depending on the speaker, e.g., disfluencies, speech rate. As a result, the continuous speech will likely generate utterance transcriptions from which the detector may generate an incorrect prediction. Furthermore, persons with PD may have long pauses or other speech anomalies depending on the disease progression. The mass-spring in the Expressor component described in Chapter 5 tries to smooth these prediction errors and compensate for different emission frequencies from the ASR and predictor.

4.4.1 Materials

We used the interview documents to collect ratings of the emotional content (i.e., valence and arousal) for each sentence of the document; 2,781 sentences in total were evaluated. A subset of these sentence evaluations were used as the training targets for the model. We collected this data using Amazon Mechanical Turk (AMT). AMT workers used a Web-based application we created to indicate their perceived emotion contained in text content (see Figure 4.2). The Web application is described in Section 4.2.1. To ensure high-quality of the training data, we only used those sentences for which at least 70% of the raters agreed on the label for valence (positive or negative). This represented 996 sentences. For the current study, we use emotional valence alone for model prediction. In prior research using this same dataset [Valenti et al., 2019b], we showed that humans had more difficulty inferring arousal than valence, reducing the sentences for which there was agreement on arousal by one-third, to approximately 600.

4.4.2 Model design

We used the model design described in Section 4.2.1 which consists of two processing steps: (i) extract the topic probabilities from each document in the set (items 1 and 3 in Figure 4.1), and (ii) use these features to predict the emotion valence of individual sentences as yet unseen by the model (items 2 and 4 in Figure 4.1). Training of the LDA model and the classifier (items 1 and 2 in Figure 4.1) was done outside of the robotic architecture and the results saved to files. These were subsequently used to initialize the predictor component in the robotic cognitive architecture; in principle, training could also take place in the architecture.

We trained the LDA topic model to generate vectors with 100 features. We used a stable, widely-used implementation of the MLP classifier from Scikit-learn [Pedregosa et al., 2011] configured with two hidden layers and 50 artificial neurons to associate the features produced by the trained LDA model with the training targets collected from the human workers. Further detail on the model design and parameter selection is given in Chapter 3. As previously mentioned, predicting both valence and arousal simultaneously proved problematic because there were relatively few training examples in which there was high agreement for both valence and arousal in the same sentence. For the initial phase of the investigation, we chose to predict valence only as the state where there was high agreement among the human raters and had many more training examples. To allow the robot to credibly mimic basic human facial affect, we made the assumption that a three classes of emotional states would be needed: neutral, positive, and negative.

Using k-means clustering of the (x,y) data we collected from the AMT study, we found three classification center points on the circumplex diagram, Figure 4.5. This classification gives valence scores of -100 to -10 as negative, -10 to 25 as neutral, and 25 to 100 as positive. Upon model evaluation, we found that classification by constant valence scores of -100 to -24 as negative, -24 to 24 as neutral, and 24 to 100 as positive give the most predictive ranges to use for classification as evaluated by its F_1 score, a common measure of classifier performance. We hypothesize that users view the circumplex as a square graph, with the center as true neutral. We therefore use constant valence values away from the center to delineate classification boundaries.

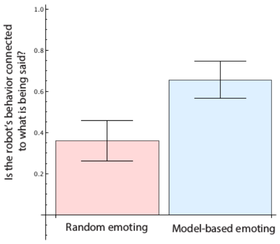


Figure 4.6: Participants first watched the video and then answered the question: "Is the robot's behavior connected to what is being said?" (*yes/no*)

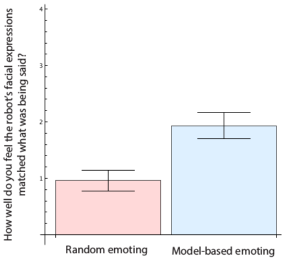


Figure 4.7: Participants answered the question: "How well do you feel the robot's facial expressions matched what was being said?" (5-point Likert scale from *Not at all* to *Always*)

4.4.3 Results and validation

When evaluating valence, the trained LDA model was used to extract 100 features for each of the 1,069 sentences in the training set obtained from the AMT workers. Then, the classifier was trained to associate the feature vectors with the values $\{neutral = 0,\ negative = 1,\ positive = 2\}$ for the emotion valence. We used Scikit Learn's *GridSearchCV* parameter sweep for tuning the MLP hyperparameters and ran a 10-fold cross validation to evaluate the model's expected performance. There were 445 examples of neutral affect, 226 examples of negative affect, and 398 examples of positive affect in our training set. We evaluated the model performance using the F_1 score weighted by the support for each class, and found $F_1 = 0.45$ where chance level is 0.33. We observed that when the model miss-classified emotion, it tended toward classifying it as neutral or positive. In a model of emotion for persons with PD, a bias towards neutral or positive rather than negative affect is desirable given that it has been shown [Takahashi et al., 2010] that PD persons often become depressed when are judged to have negative affect when they do not.

4.4.3.1 Model Validation

To validate our emotional pipeline, we conducted a study for which we recorded videos of a person talking to the camera accompanied by a robot. The person in the video was a male actor who interpreted one of the interviews produced by a person with Parkinson's disease, that was set aside from the dataset used to train the model. The interview contained an account of one enjoyable and one frustrating experience from the previous week. The actor reproduced the facial masking and affectless tone typical of Parkinson's disease. The robot either used the model to emote based on what the person was saying or emoted randomly. Each participant saw just one of these two conditions and were asked two questions (see below). Likert scale (1-5) and binary answers (which can be transformed into proportions of yes/no) were our dependent variables. Chi-square tests were conducted only on binary answers and we performed t-tests on the Likert scale variable. Although Likert scales technically do not give continuous variables, we cite, as justification, the central limit theorem which establishes that randomly generated independent observations (variables) will tend to approximate a normal distribution.

A total of 54 participants completed the study on Amazon Mechanical Turk and passed our attention checks (44.4% Female, Age Range: 21-69 years, *Mean* age = 35.02 years, *SD* = 10.53). Participants first watched the video and then answered questions about it: "Is the robot's behavior connected to what is being said?" (*yes/no*), "How well do you feel the robot's facial expressions matched what was being said?" (5-point Likert scale from *Not at all* to *Always*). When the robot emoted based on what the person was saying, using the model, 65.5% of the participants indicated the that robot's behavior was connected to what was being said, significantly more than when the robot emoted randomly, 36%, $\chi^2 = 4.685, p = 0.03$ (see Figure 4.6).

When the robot emoted based on what the person was saying, the mean participant rating of how well the robot's facial expressions matched what was being said ($Mean = 1.93, SD = 1.25$) was significantly different than the mean participant rating when the robot emoted randomly ($Mean = 0.96, SD = 0.93$), $t(52) = 3.187, p = 0.002$ (see Figure 4.7). We used the Shapiro-Wilk test to check for normality of distribution of the data in the two groups (random: $W = 0.916, p = 0.041$; model: $W = 0.989, p = 0.990$). Since the t-test is robust to small deviations from normality and because of the central limit theorem, we consider the test to accurately reflect the difference between the two groups. However, as an additional check we also performed a non-parametric Wilcox-Mann-Whitney test as an alternative, which confirmed our results: $z = 2.883, p = 0.0039$.

As an additional control, we also created a condition in which the robot did not emote at all but rather turned its head left and right randomly while the actor was speaking. An additional 25 participants completed this condition and passed our attention checks (48% Female, Age Range: 22-62 years, *Mean* age = 35.72 years, $SD = 9.75$). Significantly fewer participants (24%) saw a connection between the robot's behavior and what the actor said, $\chi^2 = 9.307, p = 0.002$.

To further explore the effects of timing we also recorded one video in which the robot used the model for emoting but the emoting was done with an average delay of 5 s. A total of 26 participants completed this condition and passed our attention checks (46% Female, Age Range: 24-51 years, *Mean* age = 31.96 years, $SD = 7.32$). When the robot emoted with delay only 50% (chance level) of the participants indicated that there was a connection between the robot's behavior and what the actor was saying, not significantly different from when the robot emoted randomly $\tilde{\chi}^2 = 1.018, p = 0.313$. Also, there was no significant difference in how well the facial expressions matched what was being said between delayed emoting ($Mean = 1.415, SD = 1.180$) and random emoting, $t(47) = 1.508, p = 0.138$. We used the Shapiro-Wilk test to check for normality of distribution of the data in the two groups (random: $W = 0.916, p = 0.041$; model: $W = 0.948, p = 0.252$). Since the t-test is robust to small deviations from normality and because of the central limit theorem, we consider the test to accurately reflect the difference between the two groups. However, as an additional check we also performed a non-parametric Wilcox-Mann-Whitney test as an alternative, which confirmed our results: $z = 1.359, p = 0.174$.

4.4.4 Discussion

Our study shows that when the robot emoted based on the model's predictions, participants recognized the connection between the robot's emotion expressions and what was being said more so than when the robot emoted randomly. Additionally, proper timing made a big difference, suggesting that in order to have the desired effect the system needs to run fast and emoting has to happen based on short chunks of speech. This further shows that other approaches such as LIWC, which might need longer chunks of text for correctly predicting emotions, would not be appropriate for embedding in an assistive robotic system meant to emote in real-time based on the person's speech content.

While this suggest that the model can be used to appropriately generate a neutral, smile, or frown affect, future work is needed to be able to automatically detect emotional content with more resolution, moving beyond these three coarse levels of valence: neutral, negative and positive. Ideally the model could be improved to be able to detect various degrees of positivity or negativity and calm or arousal. Additionally, our training and testing dataset was comparatively small. We used 448 transcripts to train the LDA model and 1,069 sentences to train the classifier, a limited amount of data compared to typical machine learning endeavors which may have thousands of training examples available. We are in the process of collecting additional labeled training examples using AMT for the purpose of developing the model's ability to predict additional emotional states beyond positive and negative. Furthermore, in order to generalize the test results to a domain beyond that described in [Tickle-Degnen et al., 2010], additional training documents from more general domains will have to be used.

4.5 Summary

We developed and evaluated an automated method for inferring the emotional valence in the continuous speech of a person with PD and embedded it in our robotic cognitive architecture. In our earlier research, we hypothesized that such a tool equipped with the ability to accurately and immediately provide feedback on the emotion content of a conversation is not only beneficial for improving the social interaction with the PD patient, it can improve the quality of life in the home. Since human emotion is communicated via multiple modalities, and through different channels, e.g., voice, facial expressions, gestures, situating such a tool in a robot that appropriately controls the robot's facial motors could compensate for the problematic visual modality of communicating emotion. We found encouraging results that showed participants in our study connected the robot's facial expressions to what was being said. Once enhanced with finer prediction resolution, a family member/caregiver could use this as a way to infer the emotional state of a person with PD. The device's technology is not restricted to the domain of PD patients; it should be able to generalize and serve as an intelligent agent useful for monitoring the emotional content of the interaction between any two parties, providing real-time feedback on the emotion valence and arousal as the interaction unfolds.

Chapter 5

A Dynamical System for Expressing Fine-Grained Emotions in a Robot

Emotions are crucial for human social interactions and as such people communicate emotions through a variety of modalities: kinesthetic (through facial expressions, body posture and gestures), auditory (through the acoustic features of speech) and semantic (through the content of what they say). Sometimes however, communication channels for certain modalities can be unavailable (for example in the case of texting), and sometimes they can be compromised, for example due to a disorder such as Parkinson's disease that may affect facial, gestural and speech expressions of emotions. To address this, we developed a prototype for an emoting robot that can detect emotions in one modality, specifically in the content of speech, and then express them in another modality, specifically through gestures.

The system comprises of two components: detection and expression of emotions, and in this paper we present the development of the expression component of the emoting system. We focus on its dynamical properties that use a spring model for smooth transitions between emotion expressions over time. This novel method compensates for varying utterance frequency and prediction errors coming from the emotion recognition component. We also describe the input the dynamical expression component receives from the emotion detection component, the development and validation of the output comprising of the gestures instantiated in the robot, and the implementation of the system. We present results from a human validation study that shows people perceive the robot gestures, generated by the system, as expressing the emotions in the speech content. Also, we show that people's perceptions of the accuracy of emotion expression is significantly higher for a mass-spring dynamical system than a system without a mass-spring when specific detection errors are present. We discuss and suggest future developments of the system and further validation experiments.

5.1 Introduction

People communicate emotion using multiple modalities. We use language, tone of voice, facial expressions and gestures to express our intentions and emotions. By looking and listening to each other, we can infer each other's emotional state and even begin to feel what the other is feeling. However, in some situations, one or more modalities may be absent, noisy, or damaged and these conditions may degrade how the human body expresses emotions. Because people rely heavily on facial expression in attributing and interpreting other's emotions and motivational states, compromised or missing modalities can deeply affect the person's ability to communicate which may lead to impaired social interactions and reduced quality of life. Such is the case for people living with Parkinson's disease (PD) which due to a condition called facial masking, are impaired in their ability to express their inner emotional state.

Our long-term goal is to develop a robot that could help people with PD express their inner emotional state and thus improve their communication with caregivers. In this chapter, we describe the emoting system which uses two components: an emotion detection component and an expression component. This paper focuses on the latter. We extended the unsupervised emotion prediction model described in

Chapter 4 and trained it to detect five different states of emotional positivity. The detected emotional state drives a mass-spring dynamical system to smooth emissions from the detection component, e.g., compensating for varying utterance frequency and prediction errors. Detected emotions are expressed as gestures in the Nao robot from SoftBank Robotics. The mass-spring also ensures that relatively more emotional "force" is needed to move a person from a more extreme emotional state than from a more neutral state. To test our system we conducted three experiments with human participants.

This chapter proceeds as follows. In section 5.2, we review prior work in applying dynamical systems to emotional modeling and discuss how emotions are embodied in humans and robots. Section 5.3 discusses the development and implementation of the mass-spring dynamical system. In section 5.4, we explain the three human-robot interaction experiments we ran to validate the full system and to highlight the contribution of the mass-spring. The results of the study show that when embedded in the Nao robot, participants recognized the connection between the robot's gestures and the speaker's utterances more when the robot emoted based on the model's predictions than when the robot emoted randomly. Additionally, the mass-spring dynamical system led to greater perceived association between the robot's gestures and the emotional content of the speaker's utterances than a model without a mass-spring element, when the emoting was done at a low-frequency (for every third utterance). This indicates that the mass-spring dynamical system is more robust to errors. Finally, in sections 5.6 and 5.7, we discuss the advantages, disadvantages, and limitations of this approach and further improvements that can be made to the system.

5.2 Background

5.2.1 Dynamical Systems of Emotion

Prior research suggests physical models, in particular the mass-spring, can be used to simulate physical human movement and create "plausible" behaviors [Fdili Alaoui et al., 2014]. Given their capacity to simulate physical behaviors, mass-spring models can also be used to simulate human movement qualities. Thus we have chosen to use the mass-spring model to simulate human gesture movement when transitioning from one emotion to another. Without this component, these movements would be completely discrete, and would not give rise to the continuous transitions between emotion states that we observe in humans. The goal of this component is to initiate behaviors in the Nao which imply transitions through intermediate emotional states on the way to more extreme goal states, or on the way back to neutrality from these extremes.

Previous studies on dynamic emotional models have produced conclusions which support the presence of continuity. Research by Xiaolan et al. [2013], for example, models emotional transitions according to the probabilities that arise in transition matrices between emotion states. The results of this study show that in a given emotional state, some subsequent states are far more likely than others. Specifically, under positive external influence, [Xiaolan et al., 2013] claims that negative emotions are most likely to transition to neutral emotions, and neutral emotions are most likely to transition to positive ones. In a discrete model based purely on the affect of human speech, these necessary intermediary steps are skipped and negative states may be forced to transition directly to positive ones. Similarly, an adaptive emotional model produced by Han et al. [2013] was developed by mimicking the emotional behavior of a human agent over the course of a conversation spanning seven emotion-specific dialogues. When charted along the two-axis Circumplex model, it is clear that their modeled agent undergoes continuous, incremental progression on its way to occupying the appropriate emotional region during each of these dialogues.

In addition to producing the intermediate steps suggested for an emotional transition by the research mentioned above, the mass-spring model for ensuring continuity supports the theory of emotional decay proposed by Velsquez [1997]. According to this theory, an emotional state is not maintained for the exact duration of the triggering stimulus and dropped when the stimulus disappears. Rather, the onset of a stimulus instigates the development of an emotional state, which slowly reverts to neutrality over time. The application of this theory can also be observed in the model developed by Yang et al. via a different implementation than the one proposed here [Yang et al., 2012].

5.2.2 Emotional Embodiment in Humans and Robots

The challenge is to design the robot's behavior so that it expresses the detected emotion in a naturalistic way which compensates for the lack of facia and gestural affect cues in the persons with PD. Research has shown that humans can successfully estimate an agent's emotional state purely from their body movements [de Gelder et al., 2015]. For this study, we used the Nao robot to display the emotional state inferred from continuous speech transcribed in real-time through manipulation of its body movements. As discussed below, a number of studies regarding human affective body language recognition as well as robotic affective body language production have helped ground our approach.

Many recent studies regarding emotional body language fall into two categories: attempts to create models by which robots are able to estimate the emotional state of a human agent through visual tracking, and attempts to assess how the same neutral action is executed differently depending on the affect of the agent. In both cases, the emotions examined were often a specific subset of the emotions described by Russell's Circumplex model, such as Fear, Anger, Joy, Excitement, and Sadness [Russell, 1980]. As will be described in a later section, we chose to represent only the valence axis of emotion, so the gestures designed are meant to represent gradations of positivity or negativity, rather than specific emotions such as these. For this reason, the poses and movements observed and produced in these other studies could not be replicated directly, but do serve as the basis for our present behaviors. In their research, Shan et al. use the FABO video database to develop an algorithm by which to assess human emotion from body gestures. This research provides evidence for the recognizability of raised arms and hands as an indicator of joy and excitement [Shan et al., 2007]. Another study, which used the same Nao robot used in our study, confirms the recognizability of happiness through raised hand and arm gesticulation [Park et al., 2010]. Similarly, research by de Silva and Bianchi-Berthouze analyze the salience of various body-feature point collections as a method of quantitatively describing body language in order to develop a classifier for emotionally labeled body language performed by an actor. This data confirms the recognizability of gestures such as drooping chest and raising hands towards one's face as an indicator of sadness [De Silva and Bianchi-Berthouze, 2004]. These studies provided static poses which could be generalized, and converted into animated body movements, to emulate varying degrees of positivity.

Other studies focused on how specific poses or motions can be altered to convey a given emotion. One such study experimented with how the upward or downward angle of the robot's head can help clarify the emotion it seems to display. The results of this study suggested that a down-turned head and face helped convey fear and sadness, while an upturned head and face helped convey pride, happiness and excitement. This data manifests itself in our gestural design through an increase in head angle change in the appropriate direction as an indicator of increased negativity or positivity. Amaya et al. [2000] developed an algorithm for determining the physiological effect of performing the same action (e.g., drinking from a cup or kicking a ball) while conveying neutral, sad or excited affects. They determined that when excited, an action is generally performed with higher joint velocity and more direct movements, whereas when sad, the joints involved often move slower and less efficiently.

5.3 Development of the Mass-spring Dynamical System

5.3.1 System Input From the Speech Recognizer

In Figure 5.1, we show the major building blocks of the robot's emotional regulation and natural language understanding systems of the DIARC [Scheutz et al., 2019a] cognitive robotic architecture. The figure shows how we supplemented DIARC with an emotion pipeline consisting of two components, the *Detector* and the *Expressor*,

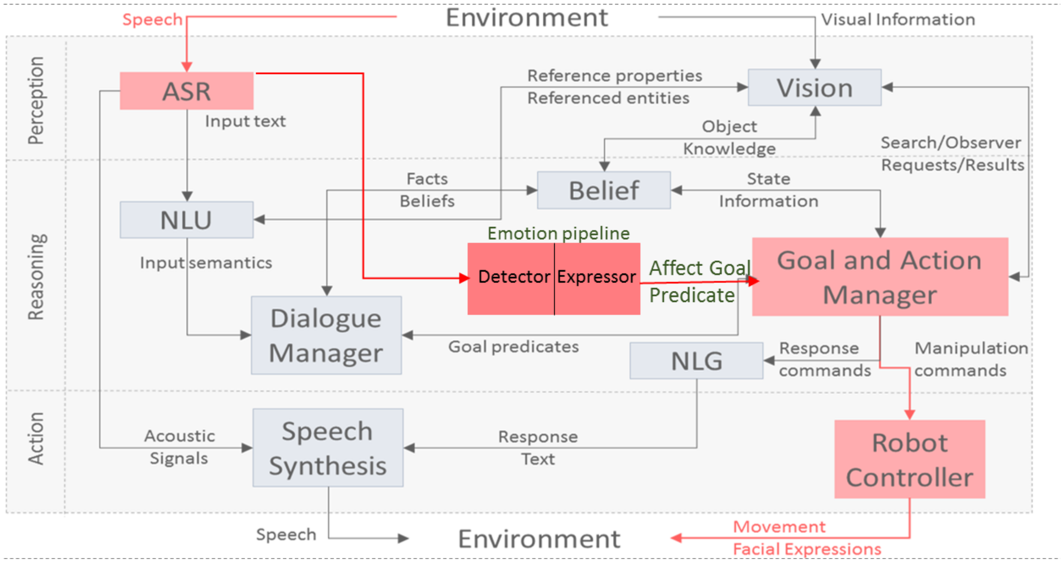


Figure 5.1: Within DIARC, the Emotion Pipeline consists of the prediction (Detector) and dynamical system (Expressor) components. It receives utterances from the Automated Speech Recognizer (LVASR) and sends its predicted affect to the Goal and Action Manager

along with their incoming and outgoing connections. The Detector component receives a text utterance from the Large Vocabulary Automatic Speech Recognizer (LVASR) in the Perception layer and gives its prediction to the Expressor, discussed in Section 5.3.3. The Expressor component then sends the goal predicates representing the desired affective states to the Goal Manager.

The LVASR is based on the chain model developed for the ASpIRE Challenge [Harper, 2015] and trained on Fisher English that has been augmented with impulse responses and noises to create multi-condition training [Varga, 2017]. The chain model uses Kaldi, a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0 [Povey et al., 2011]. Kaldi has demonstrated low error rates in a variety of challenging acoustic environments using conversational speech [iar, 2015b, Harper, 2015].

The challenge for the LVASR is to determine the endpoints of the utterances, in which words are connected together instead of being separated by speech codes such as pauses. Unknown boundary information about words, co-articulation, production of surrounding phonemes, and rate of speech affect performance [Gulzar et al., 2014] and the LVASR may generate textual representations that vary depending on the speaker, e.g., disfluencies, speech rate. As a result, the continuous speech will likely generate utterance transcriptions from which the detector may generate an incorrect prediction. Furthermore, persons with PD may have long pauses or other speech anomalies depending on the disease progression. The mass-spring tries to smooth these prediction errors and compensate for different emission frequencies from the LVASR and predictor.

5.3.2 Mapping System Output to Robot Gestures

In order to express the results of the emotion detector and mass-spring system in the physical world, it was necessary to define behaviors in the Nao robot which would best reflect the emotional data produced. All physical movement design took place in the Nao's companion software, Choregraphe. This software contains a method entitled *Animation Mode* which allows for manual moving and recording of each of the Nao's joints at various time steps. While the gestures defined here are generally based on observations of human gesticulation, prior research has produced specific data which was used to improve their efficacy and accuracy.

To produce reliable and easily interpretable gestures in the robot, we incorporated the prior research described in Section 5.2.2 as follows. Gestures meant to convey positivity incorporated raised arms and hands. This is supported by [Shan et al., 2007] as well as [Park et al., 2010]. Gestures meant to convey negativity include drooped chest and hands raised towards the robot's face. This is supported by research such as [De Silva and Bianchi-Berthouze, 2004]. Rotation of the head in upward or downward directions was also used to further convey positivity or negativity respectively. This feature is supported by the work presented in [McColl and Nejat, 2014]. Finally, the speed at which a gesture was performed was manipulated. In following with research presented in [Amaya et al., 2000], more positive gestures were carried out with higher joint velocity and swift, direct movements, while more negative gestures involved slower, less direct motion. Still images displaying a frame from each animation developed through this process can be seen in Figure 5.2.

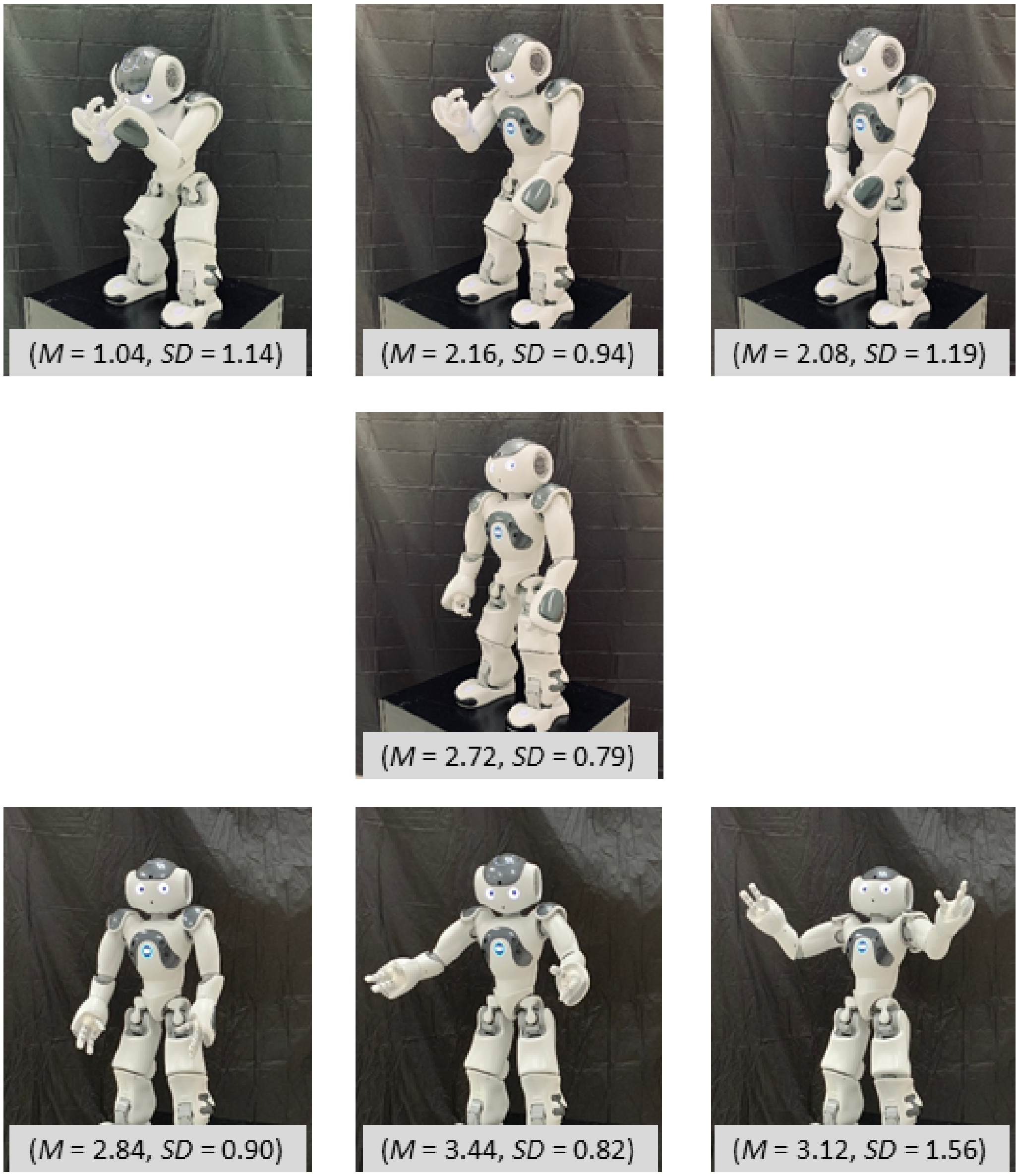


Figure 5.2: Individual gestures on the robot representing emotional valence. Top row shows three levels of increasing positivity, starting with the least positive on the left. The image in the center is neutral. The bottom row shows three levels of positivity increasing from neutral. The M mean and SD standard deviations of participant ratings are presented under each gesture.

Having designed potential gestures for the Nao to execute in order to reflect a specific internal emotional position along a valence scale, we sought to ensure that these gestures successfully conveyed their target information. To accomplish this, we carried out a validation study on these gestures in isolation from the rest of the architecture and pipeline. We recorded 10 second videos of each of the seven gestures show in Figure 5.2 which we showed to 25 human subjects (28% Female; *Mean* age = 32.08, *SD* = 9.39) who participated in the validation study on Amazon Mechanical Turk. The videos were presented in random order and four foil videos, of gestures from the standard Choregraphe database, were interspersed in between. After watching each video, participants answered the following question: "How positive or negative are the emotions expressed by the robot in this video?" The answer was given on a seven0-point Likert scale from "Very Negative" to "Very Positive with a middle point labeled "Neutral". To note that each video was rated independently, which is a more stringent test than simply ranking-ordering the videos from most negative to most positive. Means and standard deviations of ratings for each gesture are shown in Figure 5.2. When averaging across the positive gestures and across the negative gestures separately, the positive gestures are rated as expressing significantly more positive emotions than the negative ones, $t(24) = 8.90, p < 0.001$.

5.3.3 Mass-Spring Dynamical System

We have attempted to replicate the physical behavior of human expression by modeling the Nao's emotional *position* along the valence axis of emotion as a particle. In the current study, this particle behaves as the mass in a mass-spring physical system (see Figure 5.3). This physical model was selected because it aligns well with the desired behavior of this particle. In the mass-spring system, an external force applied draws the mass away from its neutral resting position and the restorative force of the spring pulls it back. Additionally, the further that the particle is from its resting position, the stronger the external force must be to increase this distance. The model we used is represented by the following second order differential equation:

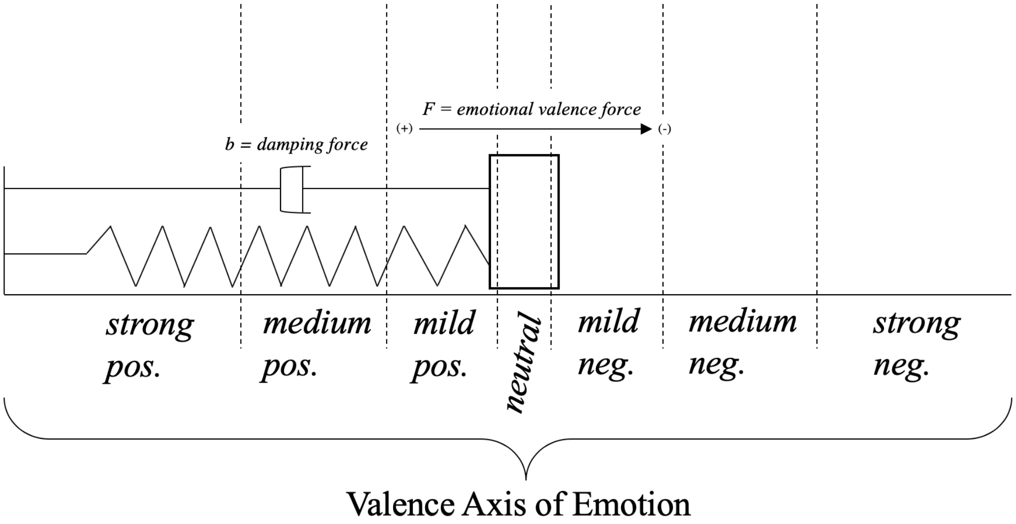


Figure 5.3: Mass-spring model with emotion particle at neutral valence. Applying a positive force F compresses the spring to move the particle toward a stronger positive valence; applying a negative force stretches the spring, moving the particle in a less positive direction. In the absence of a force, the spring's restoring effect will move the particle towards neutral.

$$\ddot{x} = \frac{-kx - b\dot{x}}{m} + F \tag{5.1}$$

where

- $\ddot{x}$ = the acceleration of the particle
- k = the spring constant, which defines how easily the spring is stretched or compressed
- m = the mass of the particle, which in the current research is set to 1
- x = the position of the particle, relative to its resting position; this is the value which is tracked internally to maintain current emotional state
- F = the force applied to the particle
- b = the damping constant applied to the velocity $\dot{x}$

The basic mass-spring model results in a system in which the particle undergoes infinite oscillation between resting position and the farthest distance attainable under a given external force. As this fluctuation does not reflect standard human emotional behavior, we incorporated a damping force into the model. The damping constant, b, minimizes this oscillation by causing the particle to lose energy as it returns to its resting point. Critical damping (b_c), in which no oscillation occurs whatsoever, is attained when $b = 2\sqrt{km}$. This is the formula used to determine the value of the b term in Equation 5.1.

A number of the parameters provided to this model, such as k, m, and b, have been determined through in-lab experimentation and tuning in order to produce consistent and expected behaviors in the system. Future research which expands on the utility of applying physical system constraints to emotional or social behavior may include a more focused study on determining the optimal parameter set for the mass-spring system to produce natural behavioral transitions.

5.3.3.1 Mapping Categorical Predictions to Force

Equation 5.1 depends on the force F which acts on the mass to move it in the positive or negative direction, depending on the valence classification of speech. The Detector component generates five possible classifications, of increasing positivity. Calling `get_prediction_value()` method of TopicModel (see Figure 5.4) returns an integer value in the range [0,4] for the predicted class (see Section 5.4.1). This integer value c_i is then used as the new input to the following Poisson-style exponential smoothing equation:

$$e_0 = c_0, t = 0 \tag{5.2}$$

$$e_t = \alpha c_{t-1} + (1 - \alpha) e_{t-1}, t > 0$$

Here, e_t represents the new emotional force. After e_t has been derived according to this formula, it is scaled from range [0,4] to range [-100,100] when applied to the mass in the physical model. The term α represents the smoothing coefficient, which defines the relative weights of the raw classification input c_{t-1} and the most recently calculated emotional force e_{t-1}. For the current research α was set to 0.6, but future research in this direction might consider experimentally determining the optimal value for this constant.

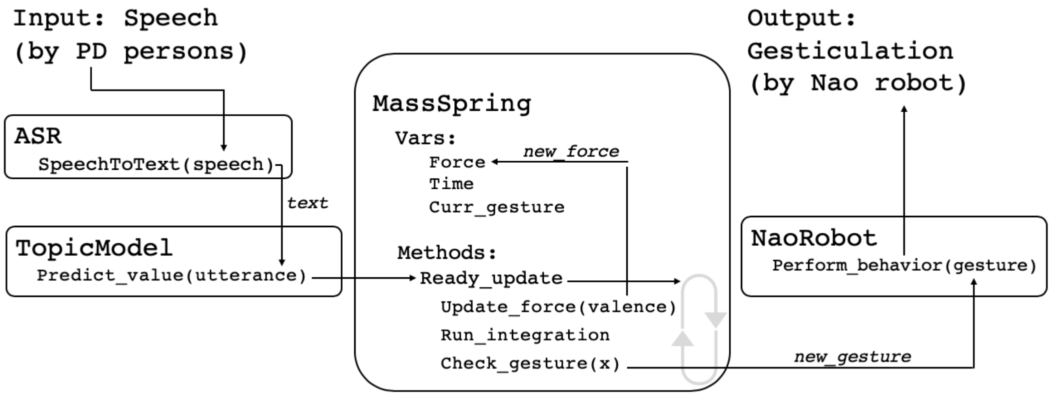


Figure 5.4: Diagram displaying the overall process at work, beginning with the speech act from a human agent and resulting in affective movement from the robot.

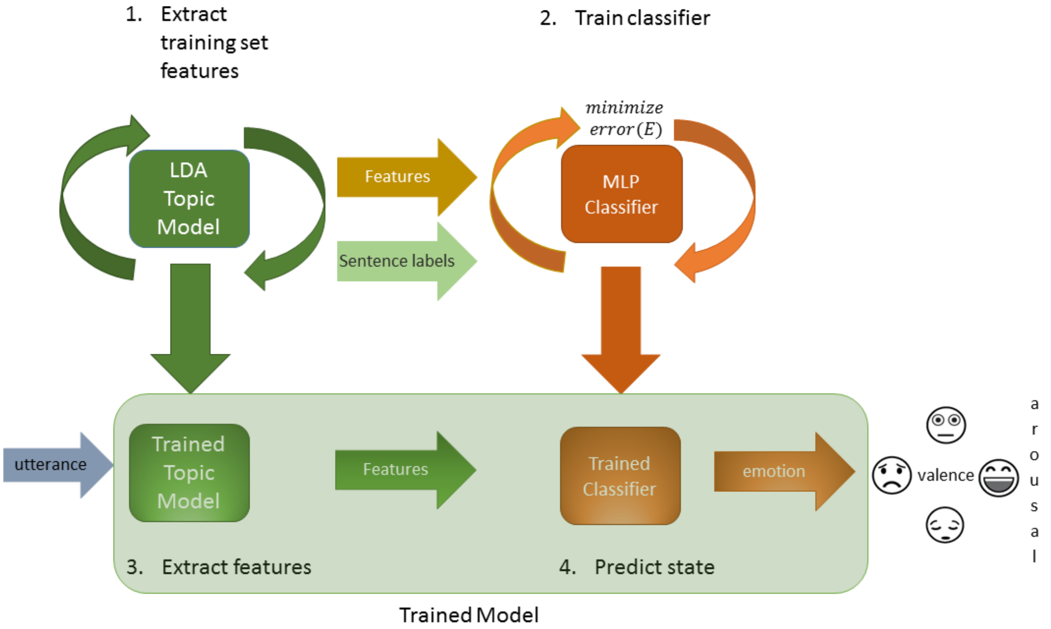


Figure 5.5: Emotion detector. (1) Document topic structure built (2) For each sentence, topic probabilities are extracted & used to train the classifier (3) Features are extracted from the utterance by the trained LDA model and (4) presented to the trained classifier to predict emotional state

5.3.4 Implementation

5.3.4.1 Emotion Detector Component

We enhanced the three-state sentiment detector described in Chapter 4 to infer five states of emotional valence {strong negative, medium negative, neutral, medium positive, strong positive}. Socher et al. [2013] reported that five levels of positivity was sufficiently fine-grained to capture the continuous values human evaluators reported and this informed the number of states we chose to detect. Our validation suggests that humans do indeed distinguish the different categorical levels of positivity expressed through the mass-spring component which gives the appearance of smooth transition from one emotion to another.

The model consists of two processing steps: (i) extract the topic probabilities from each document in the set (items 1 and 3 in Figure 5.5), and (ii) use these features to predict the emotion valence of individual sentences as yet unseen by the model (items 2 and 4 in Figure 5.5). Training of the LDA model and the classifier (items 1 and 2 in Figure 5.5) was done outside of the robotic architecture and the results saved to files. These were subsequently used to initialize the predictor component in the robotic cognitive architecture; in principle, training could also take place in the architecture.

In Chapter 3, we demonstrated that Latent Dirichlet Allocation (LDA) has been shown to be effective for inferring affect in conversational speech (also see [Shah et al., 2013, Shah et al., 2015]). We used the Gensim [Řehůřek and Sojka, 2010] implementation of LDA as the feature extractor of our model; we used the default hyper-parameter values. The generative model assumes a number of topics over which an initial distribution of documents is estimated. For this implementation, we set the number of topics to be 100, as described in Chapter 4.

We used a multi-layer perceptron (MLP) with two layers of 50 artificial neurons each; these values were selected based on a parameter sweep using Scikit-learn's GridSearchCV method. We trained the model using the *tanh* activation function with a constant learning rate with initial value 0.001 and adaptive moment estimation (i.e., “Adam”) as a fast optimizer. We used a stable, widely-used implementation of the MLP classifier from Scikit-learn [Pedregosa et al., 2011]. We trained the LDA model and the classifier outside of the robotic architecture and the results were saved to files. The Detector component used these files to instantiate the trained model in the robotic cognitive architecture.

5.3.4.2 Model Training

Training input to the model used the dataset described in Chapter 4. This consisted of individual sentences drawn from 448 documents with an average word count of 258 words, the largest containing 1,732 and the smallest, 2. The documents were constructed from selected interview transcripts from 106 male and female participants with PD, living in the community, who participated in a study [Tickle-Degnen et al., 2010] which asked them to recall two types of experiences they had during the past week: a frustrating one and an enjoyable one. The robot running the emotion detection model could then be expected to accurately predict emotion from utterances spoken in a similar contextual domain.

Ground truth labels for our model were obtained as follows. Two-dimensional (valence,arousal) emotion values for each sentence of the dataset were generated by human evaluators who used a Web-based implementation of the Circumplex model of emotion [Russell, 1980] as described in Chapter 4. In the Circumplex, valence and arousal can range from -100 (most negative/calm) to +100 (most positive/aroused) with 0 considered to be neutral. In that study, the researchers found inter-rater reliability for arousal to be low and therefore used only valence for model training and prediction. The human evaluators (N = 1,058) rated 439 documents of various lengths (269 describing a frustrating experience and 170 describing an enjoyable experience) for a total of 7,713 sentences. Each document was rated by at least four evaluators who rated between two and four documents each depending on the length. We used human evaluators drawn from the general population rather than, for example, asking PD persons themselves to label the data in some assisted way or ask some PD experts to conduct the task. The reason for this is as follows.

Correctly detecting the emotion of PD persons is a challenge because their facial expression do not match what they convey through words. Often, a person with PD can have an angry or apathetic-looking expression even when they are talking about joyful experiences. Research has shown that even specialists have a very hard time inhibiting their incorrect impressions of the person with PD when faced with dissonant emotion expressions across channels [Tickle-Degnen and Lyons, 2004]. When evaluating the emotion expression in the unaffected channel alone (content of speech), people should have no issues with detecting the correct emotion. This is exactly what our raters did: they read text transcriptions of interviews conducted with people with PD - they never saw the facial masking in conjunction with what was being said.

Using k-means clustering of the (x,y) data we collected from the AMT study, we found five classification center points. This classification gives valence scores of -100 to -10 as negative, -10 to 25 as neutral, and 25 to 100 as positive. Upon model evaluation, we found that classification by constant valence scores of -100 to -24 as negative, -24 to 24 as neutral, and 24 to 100 as positive give the most predictive ranges to use for classification as evaluated by its F_1 score, a common measure of classifier performance. We suggest that users view the Circumplex as a square graph, with the center as true neutral. We therefore use constant valence values away from the center to delineate classification boundaries.

5.3.4.3 Expressor Component

In order for the mass-spring to operate in real-time as the robot engages in a human interaction, we built a Java component which uses a second order integrator implemented by the Apache Commons Ordinary Differential Equations (ODE) package [The Apache Software Foundation, 2019]. Figure 5.4 displays the control and data flow for this entire process. The component is constantly running in parallel with the other processes in the pipeline. A method within the mass-spring, named `Ready_update` is called with rapid frequency by the scheduler of the cognitive architecture. Each time it integrates Equation 5.1 over a given time step, it first updates

F to reflect the current force derived from the most recent emotional valence prediction received from the Detect component. Each time a prediction is generated, this force attribute is modified accordingly for six seconds before reverting to 0. It then saves the state of the particle at the end of this time-step, and begins the next time-step from this saved state. This state data includes x, the position of the particle, which is used to determine which gesture to send along to the Nao for execution.

As is described in the previous section and displayed in Figure 5.2, the Nao robot is able to produce seven different gestures, intended to be dispersed evenly along the valence axis from extreme negativity to extreme positivity. In order to translate from the x produced by the integrator into an embodied behavior, we defined numerical thresholds between each gestural space along this axis, derived through observation of the distance from neutral that the particle reached during the application of various positive and negative forces. At each time step, after integration, this component assesses whether the particle has crossed a threshold from one gestural range into another, at which point the robot is instructed to switch from one behavior to another. Each behavior is designed so that it may be repeated continuously for the entire time that the particle occupies the corresponding region. Whenever this position crosses one of the predefined thresholds into a region associated with a different body language behavior, the new desired movement is reported back to the Goal and Action manager component of the cognitive architecture, which relays it to the Robot Controller Nao component for execution.

5.4 Methods

5.4.1 Design and Procedures

To validate our system we conducted three on-line experiments. The procedures were the same for each of the experiments, but the robot emoting was varied as explained below. In each experiment we asked participants to watch a video of a person being accompanied by an emoting robot (see still shot in Figure 5.6).

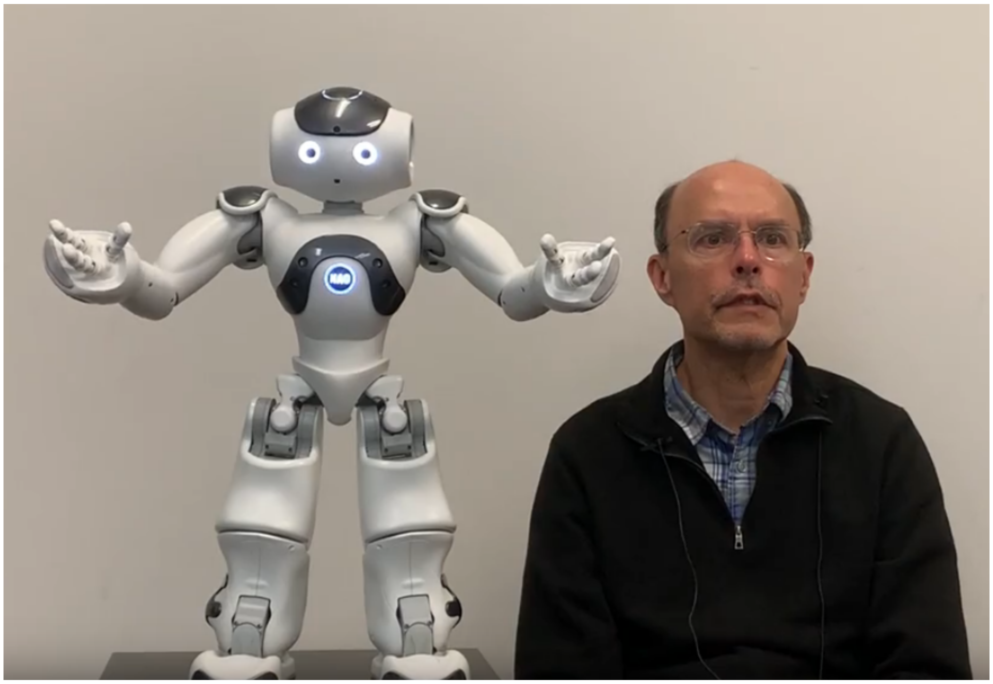


Figure 5.6: Still photo of Nao robot and PD person (i.e., a confederate) used in online evaluation of model.

We conducted the experiments on-line and used videos because we wanted, for a fair comparison, to keep constant what the person was saying across conditions. This would have been problematic in a natural interaction scenario between the participant and the assisted person. Moreover, an in-person interaction presented ethical concerns: using a confederate that actually suffered from PD for testing the robot at this stage of prototyping would have put unnecessary burden on someone vulnerable from a health-perspective, and would have created potential for stigma, while using an actor to impersonate someone with PD mimicking all motor aspects of the disorder would have constituted deception of the participant much beyond what was needed for the purpose of the validation of the system. We thus opted for a video in which a male actor speaks directly to the camera reproducing the facial masking and the affectless tone of voice that is typical of PD. The script was extracted verbatim from an actual interview with a person with PD who was talking about one enjoyable and one frustrating experience they had had the previous week. This interview was set aside from the dataset that was used to train the prediction model.

The emoting robot's behavior was varied across experiments and conditions in the following way: in Experiment 1 we compared the robot emoting based on the model described above (including the Detector and Expressor components) with a video in which the robot was emoting randomly. The purpose of this experiment was to obtain a baseline of how much people associate the robot's gestures with the person's speech. We hypothesized that a higher association to the content of speech would be perceived for the model-based emoting an the random emoting. In Experiment 2 we compared the robot emoting based on the mass-spring model described in this paper, to the robot emoting using the model but without the mass-spring element. The robot produced gestures corresponding to the emotional content of every utterance made by the person. We hypothesized that there would be no differences between the mass-spring and no mass-spring models when the emoting frequency was high (for every utterance). In Experiment 3 we used the same comparison as in Experiment 2, but this time the robot was emoting at a low frequency, only expressing the emotional content for every third utterance made by the person. This experiment was meant to showcase the robustness of the mass-spring model. We hypothesized that when the frequency of emoting drops, which might happen due to failures of the LVASR or of the Detector component, the mass-spring model would compensate for these errors. By returning to a neutral state instead of perseverating on one particular gesture, it would improve the perceived emoting accuracy. In all experiments participants were given the following instructions: "In this video you will watch a person being interviewed about some enjoyable and some frustrating experiences he's had in the past week. Accompanying him is his assistive robot. Please watch the video carefully. You will be asked questions about the person and his assistive robot."

The study used a between-group design, each participant being randomly assigned to one of the conditions mentioned above. After watching the video, participants answered the following questions: "Is the robot's behavior connected to what is being said?", which participants answered with either "yes" or "no"; and "How well do you feel the robot's gestures matched what was being said?", which participants answered on a five-point Likert scale from “not at all” to “very much so”. Likert scale (1-5) and binary answers (which can be transformed into proportions of yes/no) were our dependent variables. Chi-square tests were conducted only on binary answers and we performed t-tests on the Likert scale variable. Although Likert scales technically do not give continuous variables, we cite, as justification, the central limit theorem which establishes that randomly generated independent observations (variables) will tend to approximate a normal distribution.

Additionally, they answered further open-ended and multiple-choice questions about the robot’s behavior and the person in the video, but analyses of those answers are beyond the scope of this paper. A total of 161 participants completed the study on Amazon Mechanical Turk and also passed our attention checks (42.6% Female, *Mean* age = 37.4 years, *SD* = 11.85). The research was approved by the university’s Institutional Research Board (IRB), and participants were compensated with $1.00 USD for their time.

5.5 Results

5.5.1 Experiment 1: High-Frequency Model Emoting/Random Condition

When the robot was emoting based on our model, 80.7% of the participants indicated that the robot’s behavior was connected to what was being said, significantly more than when the robot was emoting randomly, 41%, $\chi^2(2) = 8.86$, $p = 0.003$ (see Figure 5.7). Also, when the robot was emoting based on the model, the mean participant rating of how well the robot gestures matched what was being said ($Mean = 1.96, SD = 0.87$) was significantly higher than the mean participant ratings when the robot was emoting randomly ($Mean = 1.14, SD = 0.91$), $t(53) = 3.41, p = 0.001$ (see Figure 5.7). We used the Shapiro-Wilk test to check for normality of distribution of the data in the two groups (random: $W = 0.955, p = 0.653$; model: $W = 0.999, p = 1$). This suggests that people understand the robot’s emotive gestures as related to the content of the person’s speech.

5.5.2 Experiment 2: High-frequency Model Emoting/No Mass-spring Condition

In this high-frequency emoting comparison between our mass-spring model and the same model without the mass-spring element, we found no differences between the robot emoting based on the mass-spring model and the robot emoting without the mass-spring element.

There was no significant difference between the two conditions in terms of perceived connection between the robot's gestures and what was being said (mass-spring:76%, no mass-spring: 74%, $\chi^2(2) = 0.03$, $p = 0.868$), or between the mean participant ratings of how well the robot's gestures matched what was being said (mass spring: $Mean = 1.92, SD = 1.04$, no mass-spring: $Mean = 1.74, SD = 1.10$), $t(46) = 0.58, p = 0.559$. We used the Shapiro-Wilk test to check for normality of distribution of the data in the two groups (no spring: $W = 0.966, p = 0.595$; spring: $W = 0.997, p = 1$).

5.5.3 Experiment 3: Low-frequency Model Emoting Condition/No Mass-spring Condition

When the frequency of emoting was low however, the robot emoting using the mass-spring model outperformed the robot emoting without the mass-spring element.

The use of the mass-spring led to a higher perceived connection between the robot's gestures and what was being said (mass-spring: 85% no mass-spring: 48%, $\chi^2(2) = 8.65$, $p = 0.003$) and participants rated the mass-spring emoting as better matching what was being said (mass spring: $Mean = 1.96, SD = 1.09$, no mass-spring: $Mean = 1.16, SD = 0.90$) $t(56) = 3.07, p = 0.003$. We used the Shapiro-Wilk test to check for normality of distribution of the data in the two groups (no spring: $W = 0.929, p = 0.043$; spring: $W = 0.939, p = 0.117$). Since the t-test is robust to small deviations from normality and because of the central limit theorem, we consider the test to accurately reflect the difference between the two groups. However, as an additional check we also performed a non-parametric Wilcox-Mann-

Whitney test as an alternative, which confirmed our results: $z = 2.808, p = 0.005$. This suggests that the mass-spring emoting model is robust to potential LVASR or recognition errors and is perceived by observers to be more accurate at expressing emotions from speech content.

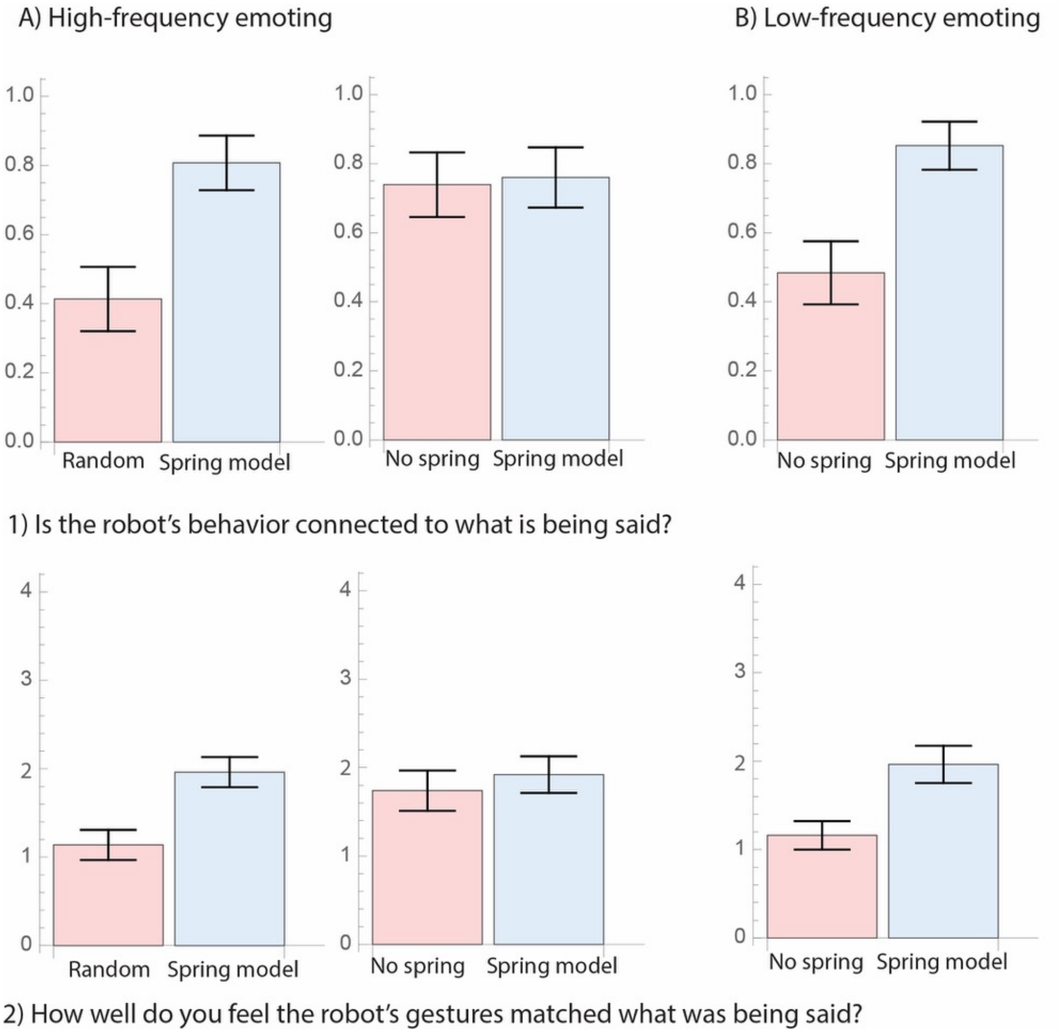


Figure 5.7: Ratings of perceived robot behavior in (A) high-frequency and (B) low-frequency emoting conditions. Top and bottom rows indicate participant responses to questions (1) and (2).

5.6 Discussion

We hypothesized that gestures could be an effective mode for conveying emotion in a robot, and that the mass-spring would be a robust design that ensures high perceived accuracy of emoting even when emotions are detected at low frequency. This might occur, for example, when utterances are spatially separated because of pauses in speech or when detection frequency is reduced to conserve power in an embedded system. We further recognized that emotion detection models are fallible and that some means would be necessary to smooth the impact of erroneous predictions on the robot's expression or when critical, emotion-bearing utterances are omitted or not recognized by the LVASR component. Therefore, we designed the mass-spring component to mediate between the prediction component and the gesture generation to serve this purpose.

Our findings support our hypotheses that humans do indeed distinguish the different categorical levels of positivity expressed through the mass-spring component which gives the appearance of smooth transition from one emotion to another. Compared to random emoting, when evaluating the overall perception of the robot's behavior and gesticulations produced by the detector and mass-spring components in concert, participants perceived a significantly higher association between the robot's emoting and the person's content of speech. The findings suggest that this system provides a suitable basis for a emoting robot companion for persons living with Parkinson's disease. Additionally, the mass-spring dynamical system led to greater perceived association between the robot's gestures and the emotional content of the speaker's utterances than a model without a mass-spring when the emoting was done at a low-frequency (for every third utterance). This suggests that the mass-spring provides necessary robustness for the emoting system, which is critical given the targeted population - people with PD often have difficulties with speech production, which might make the error rate of the LVASR component higher. Our results suggest that the mass-spring could help compensate for this.

During in-lab system testing, we noticed some delay, not exceeding 1.5 s on average between what the person uttered and the generated gesture depending on how the speech end-point was detected by the ASR component. This also impacted the accuracy of the predicted emotion. If there were disfluencies in the speech, or the person paused, the utterances would be broken up into smaller segments.

While this reduced the delay, this increased the likelihood of there being insufficient context to accurately detect the intended emotion. As noted in Chapter 3, the LDA method used in the detector has been shown to be more accurate than, for example, a sentiment lexicon approach such as LIWC in detecting emotion in short bursts of text. When evaluating the gesture set, we found that while overall the participants were able to recognize the relative positivity of the set, within the three less positive and three more positive sets surrounding the neutral gesture, there was some lack of distinction. We attribute this to having insufficient context to situate the gestures. Further research is needed to design a more definitive sequence.

5.6.1 Future Work

A future study would do well to investigate the contribution of each component of the full system in an empirical experiment under varying conditions of emotion detection frequency and error rates. Utterances containing high and low emotional variance and sequences of abrupt transitions between positive and negative emotions would further help in the analysis of mass-spring's contribution. Furthermore, human emotional state can change in far more complex ways and in more subtle gradations than the five emotional categories detected in this system. Refining the emotion detector to generate not only valence but arousal measures would could reproduce more accurately the complexity of human emotion. The challenge, then, will be to design suitable emotion expressions in the robot that reflect this complexity.

5.7 Summary

We developed and evaluated a model which detected five degrees of emotional valence in the continuous speech of a person with PD and used a spatial-temporal dynamical system to compensate for emotion detection errors and frequency of emission. We embedded the model in the DIARC robotic cognitive architecture running in a Nao robot which expressed emotion using its body movement. The system equips the robot with the ability to provide immediate feedback on the emotional state of the person with PD during conversations with their care-givers or in social-situations. Prior research has shown providing feedback on the emotion content of a conversation is not only beneficial for improving the social interaction with the PD patient, it can improve the quality of life in the home. Since human emotion is communicated via multiple modalities, and through different channels, (e.g., voice, facial expressions, gestures) situating such a tool in a robot that appropriately controls its expressive motors could compensate for the compromised vocal and facial modalities when communicating emotion.

We found encouraging results that showed participants in our study connected the robot's gestures to what was being said. Once enhanced with finer prediction resolution and further tuning of the dynamical system, the robot should be able to express emotion using any bodily movement available in a natural way and under a variety of conditions (e.g., noisy speech, rapid emotional changes). We envision that this system can be generalized to serve as a conversational agent which can monitor the emotional content between any two individuals and provide immediate feedback on the emotion content during the course of the conversation.

Chapter 6

Improving Natural Language Understanding in Spoken Dialogue Systems

Spoken dialogue Systems (SDS) are used to interact with intelligent agents through natural language. As speech input is processed, problems may arise which may cause the system to fail to generate an appropriate response. In this chapter, we show a novel framework for understanding spoken dialogue in which utterance analysis is escalated through a multi-level system *according to the feedback retrieved at the syntactic, semantic, and contextual/topic level.* Analysis is applied incrementally at each level as the system attempts to resolve the uncertainty surrounding utterance interpretation. Links to other SDS components from each of the levels can affect the agent's beliefs and conversely, other components can signal the framework to reinterpret the utterance in the context of, for example, a new topic. We demonstrate how our multi-level analysis approach can be integrated with other SDS components to improve accuracy in the SDS' ability to recognize spoken task commands. We evaluate this by comparing the semantic interpretation accuracy of utterances from two task domains given as input to an SDS, under two experimental conditions: one with the multi-level framework and one without.

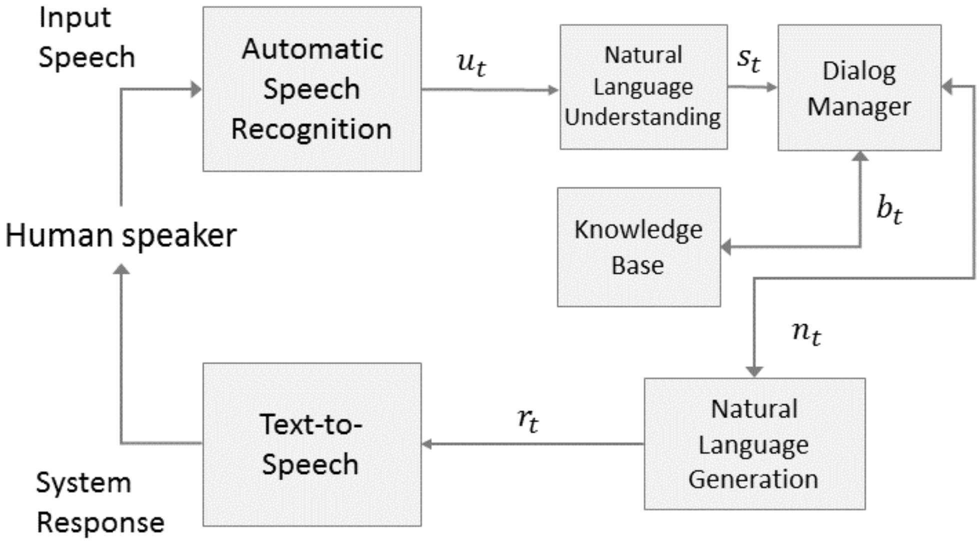


Figure 6.1: Typical components of a spoken dialogue system. At each turn t, input speech is converted to an utterance, u_t, which the Natural Language Understanding (NLU) component maps to an internal representation, s_t of the human's intent. The dialogue Manager uses this to update the agent's belief state in the Knowledge Base, b_t and then infers an natural language form, n_t from the Natural Language Generation (NLG component) which initiates a response, r_t to the Text-to-Speech (TTS) component.

6.1 Introduction

Humans use Spoken dialogue Systems (SDSs) to interact with intelligent agents using speech-based natural language [Mori, 1997, Zue and Glass, 2000, Jokinen and McTear, 2009]. Figure 6.1 shows components typically found in such systems [Scheutz et al., 2019b, Young et al., 2013]. In this example, the Automatic Speech Recognizer (ASR) recognizes the human's utterance, u_t, and sends it to the Natural Language Understanding (NLU) component. The NLU analyzes the intent and converts it to a semantic representation, s_t, passing it to the dialogue Manager (DM). The DM communicates with the Knowledge Base (KB) to send it assertions inferred from the input semantics that will be incorporated into the current belief state b_t. The DM determines its response based on the agent's updated belief state and obtains a natural language form, n_t, by consulting the natural language generation component, which then sends the response r_t) to the Text-to-Speech component.

However, problems may arise in the ASR component which can propagate through the system and cause it to fail to generate an appropriate response. For example, the ASR may recognize the word "Iraq" instead of " a rock" [Sarma and Palmer, 2004], or it may hear a novel word it has not yet learned [Scheutz et al., 2017]. Alternatively, the user may believe the system to be capable of retrieving the weather report when its domain is retrieving movie listings; in such a case, the system will need to respond to the user's out-of-domain (OOD) request [Tur et al., 2014]. Finally, in a multilingual environment the SDS may switch between different languages, e.g., a robot that a human can query in English or Japanese to initiate a Wikipedia search [Wilcock and Jokinen, 2015].

The ASR cannot recognize what it does not know about, and in the cited examples the researchers solved this problem by extending the ASR vocabulary or by adjusting the prior probability of the hypothesized word sequences. However, detecting and interpreting the user's true intention, and selecting an appropriate response given noisy human speech and ASR transcription errors, requires a method for communicating between SDS components. For example, the NLU, Knowledge Base, and Dialogue components can request the recognition subsystem to reinterpret the utterance in the event of say, a processing failure.

In this chapter, we show a novel framework for understanding spoken dialogue in which utterance analysis is escalated through a multi-level system *involving interpretation on syntactic, semantic, and contextual/topic levels* (see Figure 6.2). Analysis is applied incrementally at each level as the system attempts to resolve the uncertainty surrounding utterance interpretation. Links to other SDS components from each of the levels can affect the agent's beliefs and, conversely, other components can signal the framework to reinterpret the utterance. This may occur in the context of, for example, a new topic. To our knowledge, no other approach has demonstrated the use of such a multi-tiered system for improving accuracy in the SDS' ability to recognize spoken task commands.

This chapter proceeds as follows. In Section 6.2, we discuss prior approaches to resolving out-of-domain requests, using context to improve ASR and parser performance, and learning novel words. In Section 6.3, we place these approaches in the context of the components in our framework and show how they can be linked to existing SDSs to provide the multi-level escalation process. We also discuss how we use Topic Detection to determine context and how it was evaluated. In Section 6.4, we discuss a demonstration and evaluation of the framework using utterances drawn from two task domains in two conditions: one with the framework and one without. Finally, in Section 6.5, we discuss the advantages, disadvantages, and limitations of this approach and further improvements that can be made to the system.

6.2 Related Work

Research in improving NLU in task-oriented dialogue systems and intelligent agents can be motivated as follows. One way to ensure reliable performance of speech recognition for SDSs is to make a *closed-world* design assumption, and limit their operation to well-defined domains (for examples, see [Lane et al., 2005]). This could

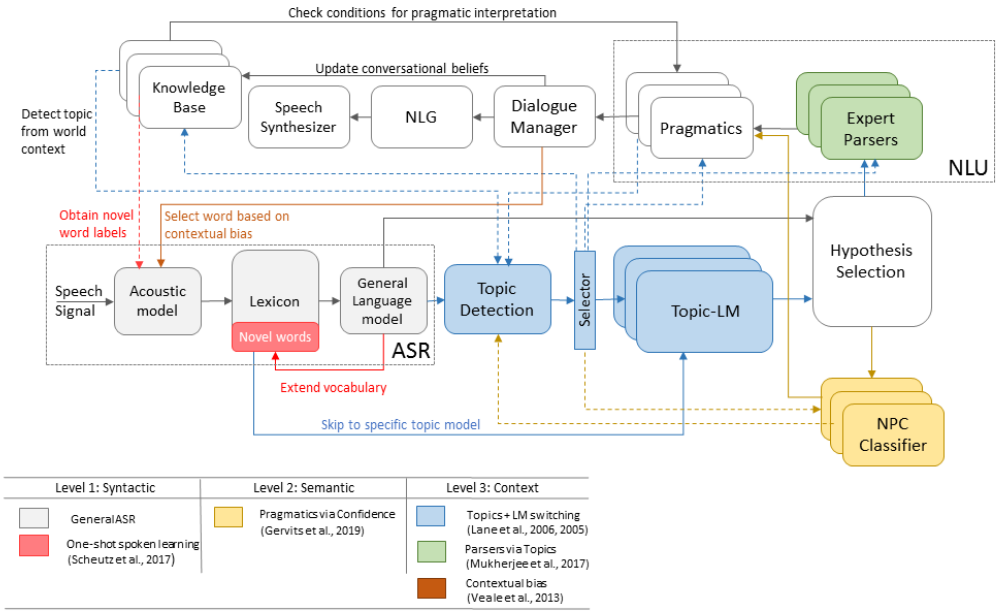

Check conditions for pragmatic interpretation
Update conversational beliefs
Knowledge Base
Speech Synthesizer
NLG
Dialogue Manager
Pragmatics
Expert Parsers
NLU
Detect topic from world context
Obtain novel word labels
Select word based on contextual bias
Speech Signal
Acoustic model
Lexicon
Novel words
General Language model
ASR
Topic Detection
Selector
Topic-LM
Hypothesis Selection
Extend vocabulary
Skip to specific topic model
NPC Classifier
Level 1: Syntactic
Level 2: Semantic
Level 3: Context
General ASR
One-shot spoken learning (Scheutz et al., 2017)
Pragmatics via Confidence (Gervits et al., 2019)
Topics + LM switching (Lane et al., 2006, 2005)
Parsers via Topics (Mukherjee et al., 2017)
Contextual bias (Veale et al., 2013)

Figure 6.2: Multi-level framework for understanding spoken dialogue encompassing prior approaches.

be accomplished by representing the dialogue model as a finite state system using a pre-defined state transition network, which assumes that the dialogue is known in advance [McTear, 1998]. This approach is not resilient to input outside the agent's domain, and so frame-based dialogue systems have been proposed. In these, the model attempts to fit the dialogue into frame slots (i.e., a "form") corresponding to an action or utterance [Xu and Rudnicky, 2000]. However, these systems struggle when utterances fail to fit into a frame.

It is desirable that the human be able to communicate in a natural and flexible manner with the agent. To enhance usability, NLU systems are built on *open-world* assumptions. In these systems, the user may provide both in-domain and OOD inputs, the latter of which may be unsupported by the system. Accepting OOD inputs could lead to errors propagating through the system, which may lead to undesirable responses unless it can reliably distinguish between the two and process them accordingly. Context detection is one approach researchers have used for OOD. Veale et al. [2013] discuss a method for applying top-down contextual bias based on the expected dialogue turn to a neural speech recognition system to improve its performance. Sarma and Palmer [2004] compute the likely contexts of all words in an ASR system vocabulary by performing a lexical co-occurrence analysis using a large corpus of output from the speech system. This is used to find the likely context for query words, and the system uses this to identify similarly-sounding, but erroneous query words.

Topic detection may also be used to infer context. Lane et al. [2006] proposed a detection framework which makes use of the classification confidence scores of multiple topics and applies a linear discriminant model to perform in-domain verification. Lane et al. [2005] describe an architecture which combines topic detection with topic-dependent language models for use in a multi-domain SDS. According to the researchers, their approach allows the user to freely switch among domains while maintaining a high-level of accuracy.

However, topic approaches use a *bag-of-words* which, along with those that are feature-based [Tur et al., 2014], have difficulty dealing with unknown words, e.g., rarely used expressions and neologisms. To overcome this problem, Oh et al. [2018] describe a method in which OOD sentences occurring in a dialogue are detected based on sentence distances. The distances are measured by sentence embedding vectors using RNN (Recurrent Neural Network) encoders and incorporate an attention mechanism.

Alternatively, Scheutz et al. [2017] describe a mechanism for detecting the intentional use of novel words in a *one-shot learning* system. Here, the ASR is modified such that when an unknown token is generated by the acoustic model, its corresponding word-level unit is discovered from the acoustic features. A nearest-neighbor classifier is used to determine whether the discovered unit represents the first member of a new word-class of the vocabulary and, if so, the class and example are added; otherwise it is added to an existing class.

In addition to using context to switch among language models, topic modeling can be applied to syntactic SDS components. Mukherjee et al. [2017] use Latent Dirichlet Allocation (LDA) to improve parser performance across multiple domains. LDA is used to find the topic structure in a document, which is a single sentence here. The sentence is assigned to the most likely topic and an “expert” parser for the topic is trained for syntactic analysis.

For situations where the domain is constrained, yet the user will be using natural language with its attendant disfluencies and irregularities, the ASR is likely to not recognize domain-specific commands. For this type of system, Leuski and Traum [2010] describe a statistical classification component which, in order to automate natural and flexible human-agent dialogue, estimates semantic meaning if the precise meaning cannot be found.

Finally, Chen et al. [2013] describe how information from multiple non-ASR components in their conversational spoken language translation system can be combined with strong baseline ASR error detector features and used to improve overall ASR error rate. The system contains built-in error detection modules that pinpoint regions in the input where the ASR is likely to fail, including a confidence estimator of the language translation (i.e., English-Iraq). Interestingly, the researchers also incorporate, as an input feature, the posterior word probabilities returned from a named entity detector to improve out-of-vocabulary word recognition.

In this section, we reviewed literature representing the main approaches to resolving OOD inputs and improving ASR performance. The contextual approaches (e.g., topic modeling, word co-occurrence, statistical classification) have the effect of changing the prior probabilities of the trained ASR by making a selection from multiple language models [Mukherjee et al., 2017, Lane et al., 2006, 2005, Sarma and Palmer, 2004], biasing the ASR word hypothesis [Veale et al., 2013], or discriminating among similar interpretations [Leuski and Traum, 2010]. An alternative approach is to extend the ASR vocabulary when a novel instance of a word class is detected as in [Scheutz et al., 2017]. Chen et al. [2013] used a combination of the two approaches. In the following section, we will discuss how these approaches have been integrated with some, but not all, of the components of the SDS framework.

6.3 Multi-level Framework

Our framework consists of three possible levels of utterance interpretation: syntactic, semantic, and context. The purpose of the first level is to select an interpretation of the user intent using the exact *syntactic* form of the utterance. The second level assumes that the utterance may or may not fit into an expected syntactic format, but that the semantic meaning of the user's intent can still be identified. In this case, a classifier is used to generate the most likely *semantics* based on previous, similar utterances. The classifier returns a similarity score to allow the Hypothesis component to select between the semantic form produced by the classifier and the semantic form produced by syntactic analysis. At the third level, the *context* of the utterance is used to restart and inform connected components to reinterpret the utterance using, for example, a new language model, classifier, or parser.

In the legend of Figure 6.2, we situate selected prior work in the framework, assigning them to the syntactic, semantic, or context levels in accordance with their approach to improving SDS performance. We place Scheutz et al. [2017] "One-shot Spoken Learning" in the syntactic level as they assume an unrecognized speech token may be a novel word. After a pattern analysis of the acoustic features, their system attempts to place the new token in the vocabulary. This flow is shown by the solid red connections in the figure. However, to recover the label of the word so that the natural language generation (NLG) can say it back using the agent's speech apparatus, the phonemic sub-units within the word feature must be recovered and mapped to the pronunciation dictionary. The dotted red line indicates the required connection for this capability.

We situate the NLG system described by Leuski and Traum [2010] in the second, semantic level as it makes no assumption that the syntactic form is correct. Level 2 uses the NPC Editor statistical classifier to generate multiple similar interpretation of the utterance, selecting the one with the highest similarity score and sending it to the Pragmatics component for intention analysis; these connections are shown by the solid gold lines in the figure. The NPC Editor classifier could ask for a back-off and reinterpretation of the utterance if the highest similarity score falls below a threshold; the connections for this additional capability are indicated by the dashed gold lines.

The basis for the context level 3 is its ability to use topic detection to infer utterance context and thus we situate the hierarchical topic classification of Lane et al. [2005] in that level. In their implementation, the researchers describe a system which can detect in- and out-of-domain utterances, and freely switch among several topic-dependent language models. In the figure, the connections and components for this system are shown in blue; however, we indicate by the blue dashed line that there could be additional connections that could further improve the interpretation. New connections from the Knowledge Base, Pragmatics, and NPC Classifier components allow them to request a back-off and reinterpretation of the utterance by inferring its context through topic detection. Connections from the selector back to those components can signal that an alternative, topic-dependent classifier, pragmatics, KB model should be used.

We also situate in level 3, the system described in [Mukherjee et al., 2017] which creates topic-specific datasets that are then used to train expert parsers. This system is shown in the green box in the figure with without a solid line connection to the Selector because the researchers have evaluated the expert parsers individually and do not specify a method for freely selecting from among language models. The green dashed lines show the connections to Topic Detection and from the Selector to indicate this added capability.

Finally, we situate the dialogue contextual bias signal system described in [Veale et al., 2013] in level 3. Rather than using a topic model to infer bias, the authors use the knowledge of common dialogue exchange patterns contained in the dialogue Manager to develop a bias signal for the ASR component (shown as a solid brown line in the figure). This is used to change the words' prior probabilities in the ASR, influencing word selection according to dialogue context. The authors describe this system as a biologically plausible cognitive model based on human perceptual decision making. As such, it provides an interesting avenue for further research into human-like ways to improve speech recognition.

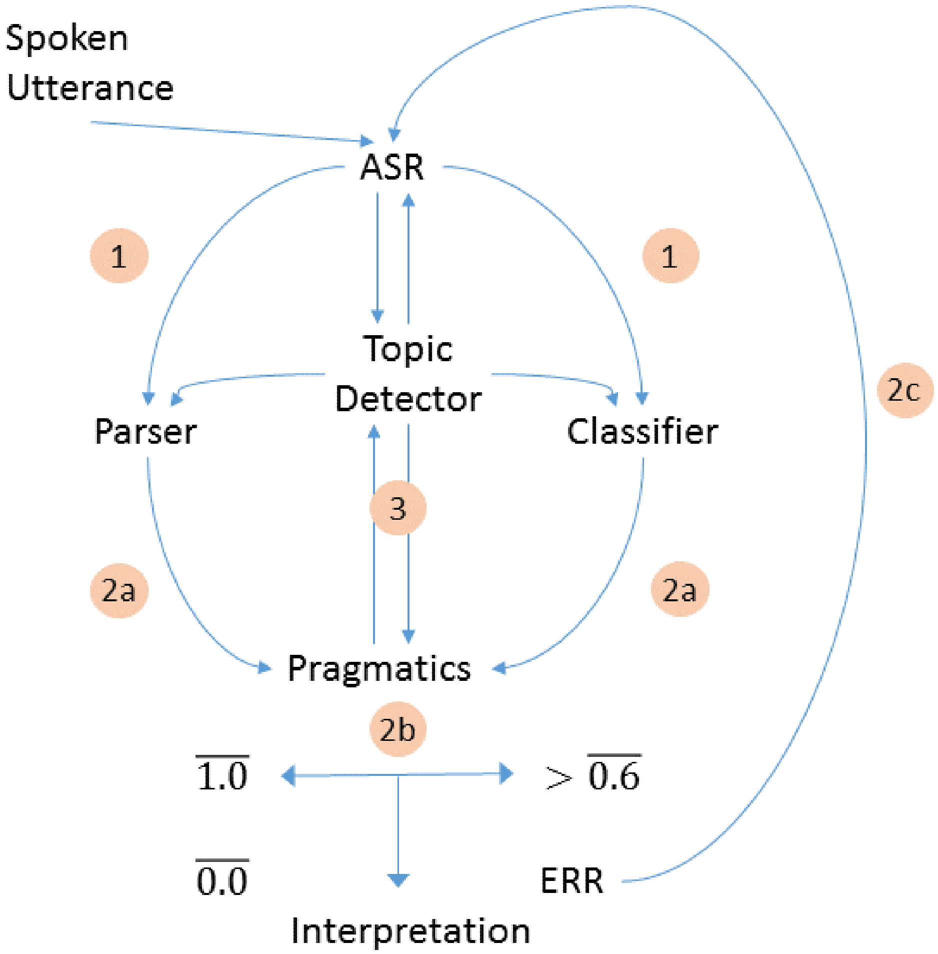


Figure 6.3: Multi-level processing flow. Level 1: The spoken utterance is received by the ASR which generates the textual utterance, passing it along to the Parser and Classifier. Level 2: the Parser and Classifier analyze the utterance semantics (2a) and send their interpretation and confidence scores to Pragmatics. Level 2b: Pragmatics selects highest score and generates the interpretation; if there is an ERR, Pragmatics requests a reinterpretation (2c) through the ASR/Topic Detector, which infers the topic and switches the LM, only if the topic has changed. Level 3: The utterance is reinterpreted through the new LM and sent to Pragmatics.

6.3.1 Implementation

We implemented the multi-level framework shown in Figure 6.2 in the DIARC robotic cognitive architecture [Scheutz et al., 2019b]. The implementation consists of the ASR, Topic Detection, Selector, Topic-Dependent Language Models, Parser, NPC Classifier components, and Pragmatics. The ASR is based on the chain model developed for the ASpIRE Challenge and trained on Fisher English that has been augmented with impulse responses and noises to create multi-condition training [Harper, 2015, Varga, 2017]. The chain model uses the Kaldi ASR [Povey et al., 2011]. For parsing, we used a symbolic, rule-based parser, and for the classification component, an implementation which is part of the NPCEditor platform [Leuski and Traum, 2010].

Figure 6.3 shows that the processing flow begins in *Level 1*, with a spoken utterance that is transcribed by the ASR component. At *Level 2a*, the utterance is sent to the parser and classifier which interpret its meaning as a semantic representation. The classifier returns all interpretations along with their similarity scores, and the classifier selects the interpretation that is above a predefined similarity threshold (0.6) and sends it to Pragmatics. The Parser, however, either successfully returns a semantic predicate assigning it a confidence score of 1.0, or fails and returns 0. In this level, simple, structured utterances (e.g., "Move to area Alpha") are processed quickly by the parser, whereas colloquial utterances such as, "Um can you like come to Alpha", can still be successfully interpreted by the classifier based on its similarity to expected utterances.

At *Level 2b*, Pragmatics selects the highest score returned by the Parser and Classifier and generates the interpretation. If the Parser fails ($score = 0$), or the Classifier cannot find a semantic interpretation with a $score > 0.6$, then it requests a reinterpretation (*Level 2c*) through the ASR/Topic Detector. If the topic has changed (*Level 3*), the Topic Detector will switch to a generic language model that is a mixture of topic unigrams, and the utterance will be reinterpreted. If there was no topic change, the framework assumes there cannot be a valid interpretation, and will generate an appropriate response to the user. The generic LM is used so that the Topic Detector has a basic utterance to which it infers a topic distribution. The Topic Detector uses the distribution to select a domain-specific LM, and the utterance is reinterpreted using the new LM, passing once again to the *Level 2* processing.

6.3.2 Topic Detection

We used LDA [Blei et al., 2003b] to infer topics in a heterogeneous collection of textual data. The intuition behind LDA is that documents exhibit multiple topics and that only a small set of topics are contained in a document and that they use a small set of words frequently. As a result, words are separated according to meaning, and documents can be accurately assigned to topics. The LDA model can function as a topic detector, which can generate the topic distribution contained in a document (which in this case is a single utterance). It detects a topic shift from one utterance to another by comparing the KL Divergence between the two topic distributions. If the difference is above a pre-determined threshold, a shift is indicated. This is used by the Topic Selection component to signal the Selector to switch to a new language model. As stated previously, this mechanism can be extended to select among alternative parsers, classifiers, knowledge base components, etc.

6.3.2.1 LDA Model Training

We used the Gensim [Řehůřek and Sojka, 2010] implementation of LDA to train the topic detector and extract the topic distribution from the utterances; we used the default hyper-parameter values. The generative model assumes a number of topics over which an initial distribution of documents is estimated. Once trained, the model infers the thematic structure of the document collection, i.e., the per-word topic assignment and the per-document topic proportion. The model can then be given a previously unseen utterance and attempt to fit it to the thematic structure of the document collection. The result of such a query is a vector representing the topic distribution t of the input utterance u i.e., $P(t|u)$. Also returned are the words w_j of the utterances that the model has assigned to each topic along with the probability of it being in topic i, i.e., $P(w_j|t_i)$.

6.3.2.2 Topic Detection and Evaluation

The topic detector is implemented as a Python function which queries the trained LDA model using utterances u_1, u_2 and returns their topic and word distributions along with a Boolean indicator of whether the topic has changed. The indicator is set by comparing the topic probability distributions of u_1 and u_2 using Kullback–Leibler (KL) divergence, a measure of how one probability distribution is different from another [Kullback and Leibler, 1951]. If the divergence falls below a pre-determined threshold, the the function returns *False*, otherwise it returns *True* indicating a topic shift. We tested the topic detector's ability to distinguish utterances drawn from one domain or the other as follows.

We used k-fold validation with $k = 5$ to train and test the LDA topic detector using the document collection described in Section 6.3.2.1. Membership of a test utterance in either of our two domains (see Section 6.4) indicates the ground truth topic membership. We prepared the folds by separating all sentences into n equal parts. For each fold, we reserved that fold as "testing", and used the others as "training" to train a new LDA topic model. The training sentences are recombined into their respective files (so that the topic model is trained with each document being an entire file). Thus, utterances are grouped together by file when training occurs. For the sentences in a testing set, at most 100 random sentence pairs are selected (e.g., [a, b, c, d] has pairs [(ab), (ac), (ad), (bc), (bd), (cd)]). Since the number of pairs increases exponentially with the size of the original list, a limit is imposed on the maximum number of sentence pairs to select. If a pair of sentences come from different files, then this is considered ground truth of a topic change, and vice versa. Sentence pairs are fed to the topic model, one after the other, and the predicted topic change is compared against the ground truth topic change. If the KL divergence of the two sentences is equal to or exceeds the input KL threshold (0.5), then this is considered a topic change, and otherwise not.

6.4 Framework Demonstration

To validate our system, we conducted an evaluation of its accuracy in interpreting natural language utterances from different human-robot tasks. The goal was to compare our multi-tiered system to one which only used syntactic parsing. Using the data collected in other human-robot interaction experiment environments, we obtained two separate corpora of natural language utterances used to instruct and otherwise communicate with a robot in a specific task environment. For this implementation, we trained the topic detector on two topics, corresponding to the domains we wish the detector to distinguish among. The first domain, *SpaceStation* consisted of a transcription of 26 participants in an experiment in which they gave commands in natural language to control several robots repairing components of a space station [Gervits et al., 2020]. Out of 663 utterances, some of which were duplicates, 363 unique utterances were chosen, out of which 50 were withheld for the test dataset and 313 were used to train the LDA model.

The second domain, *Diorama*, consisted of a transcription of 33 participants in an experiment in which participants taught new skills to a robot learner using natural language [Bennett et al., 2017]. Out of 680 utterances, some of which were duplicates, 525 unique utterances were chosen, out of which 50 were withheld for the test dataset and 475 were used to train the LDA model. The sentences from the SpaceStation domain comprised one document and those from the Diorama domain comprised another. The LDA model was trained on a union of the two document collections.

We set up a pipeline for incoming utterances wherein each utterance would be processed in parallel by two different systems: our three-tier framework which included topic-switching and a bag-of-words classifier, and a control framework which possessed only the baseline tier of interpretation through syntactic parsing.

In both the control and multi-tier systems, parser rules were written by hand based on 100 utterances from each corpus, which were also used to train the language models for ASR. For the control framework, these syntactic parsing rules were combined into one parser dictionary, whereas for the multi-tier system, the parser swaps between topic-specific dictionaries at the topic identifier component's signal. Similarly, the control system's language model was trained on the combined set of 200 utterances, while the multi-tier system contained two separate models each trained on 100 utterances from the distinct tasks. For the multi-tier system, two different classifiers were trained on the two different sets of utterance training data that had been hand-labeled with the correct semantic interpretation in predicate 1st-order logic form for each utterance.

This pipeline was fed a test set of 50 utterances from each corpus (100 utterances total) that were withheld from the training data. Relevant output such as utterance transcription, semantic interpretation, and in the multi-tier framework's case, topic identification, were logged and manually annotated by the experimenter with the ground truth values of these variables: the correct transcription of each utterance, its intended interpretation in symbolic 1st-order logic form, and the task (topic) from which it originated.

6.4.1 Results

To investigate the differences in accuracy between the transcriptive and interpretive abilities of our multi-tier framework and control framework, we compared the output of each framework per utterance to its respective ground truth value using a Levenshtein distance metric. For transcription accuracy, we found the token-based Levenshtein distance between each utterance and the ASR transcription of that utterance[1]. In the multi-level framework, the mean of this value was 1.23, meaning that the ASR transcription was off from the ground truth by a mean of 1.23 word deletions, insertions, or substitutions. The mean distance in the control setup was 1.00. Assuming the variances are homogeneous, we conducted a paired two-tailed t-test which showed no significant difference in transcription accuracy between frameworks ($t(99) = -1.707, p > .05$). We then conducted Levene's test

[1] This was token-based instead of character-based because we did not wish to reward misrecognition of a shorter word over that of a longer word, i.e., "canned" and "can" vs "canned" and "tanned"

of the homogeneity of group variances which is a stricter test when the data is not normally distributed. Since $p - value = 0.062475 > p = 0.05$, we cannot reject the null hypothesis and conclude there is no significant difference between the two group means and so the t-test satisfies the homogeneity of variance assumption.

For interpretation accuracy, we found the token-based Levenshtein distance between each utterance and the system's semantic interpretation of that utterance to measure our dependent variable: how close was the predicate for of the utterance interpretation to the ground-truth. In cases where the system was unable to come up with any interpretation, the distance defaulted to 6. In the multi-level framework, the calculated mean of this value was 1.57. The mean distance in the control setup was 4.84. Assuming the variances are homogeneous, a paired two-tailed t-test found a significant difference in these means ($t(99) = 12.425, p < .001$), demonstrating that the multi-level framework is significantly much more accurate than the control. We then conducted Levene's test of the homogeneity of group variances which is a stricter test when the data is not normally distributed. Since $p - value = 0.911839 > p = 0.05$, we cannot reject the null hypothesis and conclude there is no significant difference between the two group means and so the t-test satisfies the homogeneity of variance assumption.

The control framework was generally only able to accurately identify the semantic interpretation of the utterance in cases where the utterance matched exactly to the grammatical rules specified in the parser. In addition, it occasionally made the correct semantic interpretation in cases where it misrecognized an unexpected utterance as an expected utterance with the same meaning. For example, the utterance "hey robot one come fix this tube" was misrecognized as "fix the tube" by the control framework and thus correctly parsed with the meaning of "repairTube()".

In other cases of transcription inaccuracy, the control framework occasionally recognized portions of utterances from the wrong task corpus. For example, it recognized the utterance "robot one go to left four" as "robot one go left", thus interpreting the utterance as "move(left)". While the diorama task has a need for this level of directability in the robot's movement, the space station does not, and the action “move(left)” is used exclusively in the diorama task. However, the control framework had no reason to suppose that this command was less likely to be uttered in this context, either on a speech recognition level or on a semantic interpretation level. In all other cases, the control framework was unable to parse the unexpected utterance, even if the ASR Component transcribed it completely accurately. In contrast, the multi-level framework, even if unable to get a wholly accurate transcription, was generally able to come up with a semantic interpretation which, if not exact, was fairly close to the intended interpretation, on average only off by one or two arguments. The topic identifier in the multi-level framework also correctly identified the topic of the utterance 93% of the time.

6.5 Discussion

We hypothesized that the multi-level framework would be able to perform better overall in semantic interpretation with no detriment to ASR accuracy. The results show that there was a statistically significant difference between the semantic interpretation accuracy of the multi-level framework and that of the the control framework, with the semantic interpretation being more accurate in the multi-level framework. In addition, there was no significant difference between the transcription accuracy of the multi-level framework and the transcription accuracy of the control framework. The control framework performed very well on expected utterances (speech that matched the syntactic structure of utterances from the training set), but very poorly on unexpected utterances. Its success at semantic interpretation was binary: either 100% certainty or 0% certainty. In contrast, the multi-level framework was able to guess in uncertain situations due to the classifier, leading to far greater overall success at interpretation. Arguably, executing a wrong command could be potentially worse than not understanding a command at all. However, having an uncertain estimate of what the user wants is better than no estimate. Rather than executing the command, further error recovery could begin based on information from other components. For example, if the agent’s certainty regarding its interpretation is not above a certain threshold, its dialogue manager could initiate a confirmation or clarification request, or its knowledge database could be solicited for contextual information or dialogue history that might resolve uncertainty. This would be a direction to explore in future work.

Accuracy is improved for interpreting full user tasks. The accuracy of the human's shorter replies in response to the agent's clarification request, however, were not included. For example:

Human: "Move to alpha left one"

Robot: "Which location?"

Human: "Alpha left one" *(this short reply not included in testing)*

The primary reason is that the context of the dialogue is maintained in this architecture and as a result the Classifier might settle on "move to alpha left one" (correct) but might equally settle on "fix tube alpha left one" (incorrect) instead, having no idea that moving is preferred over fixing. Maintaining context would be a useful cue to the NLU component and would help bias toward the preferred interpretation.

We note that even in the control framework, the LM was trained on a selective portion of data only containing the two topics. In contrast, the Aspire Chain Model default LM is used for general dialogue. Thus, there is not a substantial difference in the ASR word error recognition (WER) between the two. When run with the default, the control's WER is far worse. This will vary depending on how specific to the task the utterances are. For example, utterances like "go to left four" or "drive forward pushing box c" are transcribed as the irrelevant phrases "gonna last for" and "dry forward pushing box see", while "what are you doing right now" and "knock down the yellow tower" are recognized correctly.

As mentioned in Section 6.4.1, the success of the multi-level framework depends on whether or not the topic is identified correctly. If the topic is misidentified, all other output from the system will also be incorrect. Though a 93% success rate of topic identification appears good, it is likely that this success rate would decrease as more topics are introduced. For this reason, focus should be placed on how other components of an autonomous dialogue system can be integrated into the process of topic identification, so that the burden is not solely placed on ASR. There are several examples of how this can be handled. For example, if the parser and classifier both fail to come up with an interpretation above some threshold of certainty, they could prompt the topic component to switch the topic to the second-place choice and attempt another pass at NLU, or additional information could be solicited from the system's knowledge base about the dialogue history, previous goals, goal status, or world state that may further assist with topic identification.

In Chapter 4, Section 4.2.1, we describe how topic modeling is used to detect emotion from speech. It is also possible to use the detected emotion to improve the user engagement with the SDS (for two examples, see Chapter 2, Section 2.6). The detected emotion could be stored in the Knowledge Base which can be used to modulate the agent response according to the current emotional state and how it changes over time. Not only could this lead to increased engagement with the SDS, an agent which understands the user emotional state can make the agent appear to be more emphatic, increasing trust [Cramer et al., 2010].

We used only the similarity scores from the classifier and parser to determine interpretation quality. However, it is possible to include other measures of uncertainty that are available, such as the ASR's word-level transcription confidence. This might be used, for example, ahead of the semantic components to signal an earlier switch to another language model. Alternatively, a more robust ASR error detector might supplement the ASR confidence with additional metrics such as: LM perplexity, number of competing words, acoustic model deviation from true scores, parts-of-speech, word vs. grapheme disagreement, and homophone indicator. Chen et al. [2013] used these features to predict error labels for ASR hypothesis. In addition, a more robust word boundary detection using acoustic-prosodic as described by the researchers can be used to develop a confidence in the ASR-hypothesized word boundary detection. This might have the greatest benefit for the framework when given OOD utterances. These utterances are often broken up into multiple in-domain words and thus, word insertions are frequent, making up about 40% of word errors according Chen et al. [2013].

6.6 Summary

Natural language interaction with SDS can result in errors which propagate through the components, causing the semantic interpretation to fail. We developed a multi-level framework in which utterance analysis is escalated according to feedback received at the syntactic, semantic, and topic level. We situated this framework in the context of prior research in improving speech recognition and natural language understanding and showed how they have been integrated with some, but not all, of the components of our framework. In a demonstration in which humans used natural language to initiate commands controlling robots in two separate domains, we showed how these approaches can be integrated with other SDS components. We found improved accuracy in the SDS' ability to interpret spoken task commands. By integrating multiple different potential routes for understanding into the dialogue system, we allow for better recovery across the system.

Chapter 7

Building a Bilingual Robot

As of 2018, About 22% of the population in the United States is bilingual as are other countries with a “national language” (i.e., France). The proportion of the population in many other countries have a significantly higher rate of bilingualism. A bilingual robot which could not only switch from one language to the other but “code switch” within a dialog as humans do, would be more functional in a multilingual environment.

In this chapter, we investigate and demonstrate two computational models that further our understanding of how to extend the model framework to a bilingual environment. The first is a computational model of the inhibitory control theory which states that the non-target language of the bilingual is suppressed by top-down contextual cue. This indicates the need to provide a way for our model framework to provide a similar contextual cue to control the active language. The second is a computational model of bilingual memory which describes a psychological theory of how words from the multiple languages interfere with each other and how the desired word is selected using the top-down control mechanism. Similarly, we demonstrate which parts of the model framework needs to be modified so that it can freely switch among syntactic and semantic components to correctly interpret the meaning of the human utterance.

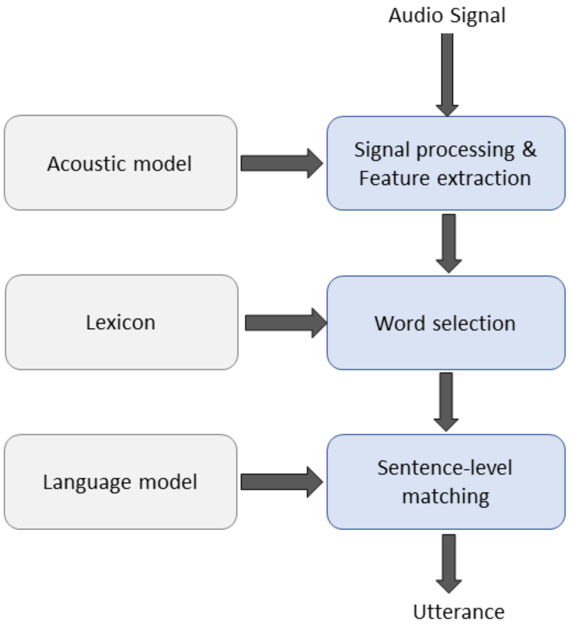


Figure 7.1: Automated Speech Recognizer (ASR) model. *Acoustic model (AM)*: Input is the audio signal and the outputs are typically acoustic feature and phonemes. *Lexicon*: also called vocabulary or dictionary. The list of words that exist in the language that the system can decode. *Language model (LM)*: or grammar, defines how words can be connected to each other. It can be defined by a set of rules, or a large list of word tuples (n-grams) with assigned probabilities.

7.1 Background and Related Work

Increasingly, our culture is becoming more diverse and bilingual. As of 2018, About 22% of the population in the United States is bilingual; this is on par with other countries with a “national language” such as France, but about one-half of a multilingual country such as Switzerland. Even so, there are U.S. cities such as Miami and L.A. with a large population of bilinguals and for these population centers, a socially assistive robot with a bilingual capability may improve user engagement. This might be the case in an elder care environment where the resident feels most comfortable code-switching among their two languages, for example.

While certain robots support multiple languages, most robotic architectures are monolingual: i.e., language understanding and production is configured for one language at a time. For example the SoftBank Pepper robot has the capability to speak 12 languages but must be configured to one language during the set-up procedure [pep, 2020]. For example, in a Robot Assisted Language Learning (RALL) on English vocabulary learning and retention of Iranian children with high-functioning autism, a SoftBank Nao robot was configured to use only English [Fdili Alaoui et al., 2014]. Wilcock and Jokinen describe a demo of their multilingual WikiTalk system in which a robot can switch languages upon prompting by a human who can then query Wikipedia the new language [Wilcock and Jokinen, 2015]. According to Laxström et al., the system uses separate speech recognition and synthesis modules for different languages [Laxström et al., 2016].

Referring to Figure 7.1 we can see some of the challenges in designing a robot to understand multiple languages at the same time. In a typical ASR design, there are usually three logical components: the Acoustic model (“AM”), the Lexicon, and the Language model (“LM”). Each component is generated during model training, the AM responsible for mapping acoustic features to phonemes, the Lexicon providing the words in the vocabulary, and the LM, the underlying language structure. A naïve implementation of a multilingual ASR would use different Lexicons, AMs and LMs for every language the ASR could understand. However, research in deep neural network multilingual and cross-lingual acoustic models attempts to make it easier to train and generate a single AM that is multilingual. While the impetus for this research is creating ASRs for under-resourced languages, these techniques can generalize to speech recognition and production for other language types. Under-resourced languages ("ULs"), for example, lack a unique writing system or stable orthography, have limited presence on the web, and lack linguistic expertise or electronic resources for speech and language processing. For an overview of the ULs, their challenges, and technological approaches, see [Besacier et al., 2014].

Research in building ASRs for under-resource languages promises more efficient ways to collect training data to generate multilingual language models. While these approaches make it easier to bootstrap new acoustic language models from common languages such as English, they still assume a lexicon and language model for the new target language; Besacier et al. [2014] discuss approaches for the lexicon and language model when the training corpus is sparse. Holzapfel discusses an approach to simplify multilingual grammar specification [Holzapfel, 2005]. He introduces grammar interfaces, similar to interface concepts used in object oriented languages, to improve compatibility between different grammar parts and to simplify development. In an example of a multilingual ASR, Barnard et al. describe an spoken dialog system which can recognize eleven South African languages with 54% - 67% accuracy. However in a query application where the vocabulary was restrict to 10 words, accuracy was expected to improve considerably, close to 90% [Barnard et al., 2010, Van Heerden et al., 2009].

The mechanism allowing the ASR to switch among the ASR components during continuous conversational speech, to allow for "code switching" (a practice in which bilinguals freely switch between language either within or between sentences [Myers-Scotton, 2006]) is an active area of investigation in speech recognition [Yilmaz et al., 2016, Vu et al., 2012]. Hence, we suggest that our robots should be trained to learn and process multiple languages simultaneously as bilinguals are thought to do. Toward this goal, we investigated creating a computational model of bilingual memory in which the language control processes adapt to the conversational context and change as the second language is acquired. There are three parts to this model which we desire to test empirically: (i) bilingual conceptual memory, (ii) word selection, and (iii) language control. Respectively, these mean: (i) how the model represents a word other words that are conceptually related to it, in both the primary and secondary language, (ii) how a word memory is "chosen" from memory, according to the language task, e.g., during speech production or reading, and (iii) how the word in the intended language is chosen to be spoken or recognized rather than its equivalent in the unintended language.

This chapter is divided into two sections. In Section 7.2, we investigate an account of language control: Green's regulatory processing model [Green, 1998]. We simulated the Inhibitory Control theory to confirm whether or not the theory can account for the language switching costs seen in an empirical experiment conducted by [von Studnitz and Green, 1997]. Data from the model simulation supported the empirical data, finding a language switching. Adjusting the connection weights on the word-level inhibitory connections alone was enough to cause the model to fit the empirical data suggesting that the inhibitory effect is primarily due to schema inhibition and not language-tag inhibition in the bilingual lexico-semantic system, for a language-specific lexical decision task, in accordance with Green's hypothesis.

In Section 7.3 we describe a model of bilingual memory and demonstrate how access is controlled base on the conversational context and speaker proficiency levels. Finally, in Section 7.4 we demonstrate which components of the model framework of Figure 7.12 have been modified to include what has been learned from the inhibitory control and bilingual memory investigations.

7.2 Language Control

Over the years, a general consensus has emerged in bilingual research that both of a bilingual speaker's languages are active simultaneously and that there is a lexicon unified across both languages. Experimental results have repeatedly shown that the bilingual speaker's two languages compete in terms of phonological representations (i.e., accents) and word meaning, but proficient bilingual speakers rarely confuse words from the competing language and, despite the intrusion of an accent, can usually be readily understood. Thus, there must be a cognitive process that allows bilinguals to control what they are saying and understand what they are hearing, when they are speaking, reading, or listening in the target language.

One theory proposes the existence of a "language switch" which, when set in the correct position, effectively blocks the other language [Macnamara and Kushnir, 1971]. An implication of the language switch theory is that there would be a cost of switching between languages (i.e., the cost of "throwing the switch") and, moreover, that the switching costs between L1 and L2 would be symmetrical (due to the very nature of a "switch"). However, asymmetrical costs were found in an experiment by Kroll and Stewart [1994], thus prompting the question of what cognitive process might account for this asymmetry in switching costs? The Inhibitory Control (IC) hypothesis of Green [1998] attempts to explain this asymmetry by proposing that there is an increase in time needed to resolve competition among activated word forms (i.e., lemmas) in L2.

In our investigation, we set out to provide evidence for the IC hypothesis by constructing a computational "proof-of-concept" model that implements the hypothesized inhibitory mechanism in the context of a lexical decision task. We start by explaining the IC hypothesis and discussing some of its predictions as they relate to this paper. Next, we review von Studnitz and Green [1997] study, the empirical data used, and the experiment's procedure. We then introduce the model framework used to construct a computational simulation of language switching predicted by the IC hypothesis, and report the model's results, comparing them to the empirical data and discussing the model's advantages and disadvantages. Finally, we point to areas in which the model might be extended in order to account for more of the effects reported by Von Studnitz and Green and how the model might be generalized to language switching in speech production.

7.2.1 Language Control in Bilinguals

In the IC hypothesis, a set of language-specific processes and language task schemas, operating under the control of a general cognitive supervisory process reactively inhibit competitors at the lemma level of the lexico-semantic system using its language tags. A lemma is a representation in the lexico-semantic system that contains syntactic information which Green [1998] identifies as the locus of language membership. IC extends the Kroll and Stewart [1994] revised hierarchical model (RHM) which proposes that a bilingual's first and second languages (L1 and L2) are connected bi-directionally through links whose strengths vary as a function of the language. However, this model has some limitations. For example, RHM does not specify how a bilingual engaged in a language translation task avoids naming the word to be translated and the IC model suggests a plausible mechanism. Kroll and Stewart [1994] found that when asking individuals to translate words that were blocked by category, for forward translations (i.e., L1 to L2) participants took longer to translate those words than when they were randomly presented. No such effect was observed for backward translations (i.e., L2 to L1). This suggests that in forward translations, according to the researchers, [Kroll and Stewart, 1994, 168], blocking words by category activates the conceptual element, creating difficulty in selecting a single lexical entry for production. Green [1998, 73] hypothesizes that there is an increase in time needed to overcome competition between L2 lemmas which have become activated and suggests the presence of a control mechanism to account for the observed effects.

Green [1998] develops the idea for a control mechanism by building upon the observations of Grosjean [1997] that bilinguals operate in different language modes. They may be speaking in L1, L2 or, in the appropriate context, mixing both their languages. Green hypothesizes that there must be a regulatory mechanism that is both sensitive to external input and has the capacity for internal control. Building upon Green's prior research derived from the "contention scheduling model" proposed by Norman and Shallice [1986], Green developed the IC hypothesis. Norman and Shallice argue that most attentional conflicts occur in the initiation of an action rather than its execution and propose a two-level control mechanism. The first level is a contention scheduling process that selects from competing schemas; the second is a supervisory attentional component that oversees and biases the selection process. Incorporating this theory, IC hypothesizes that the intention to perform a specific language task is executed by a supervisory attentional system (SAS) which affects the activation of language task schemas that are themselves in competition to control the output. Thus, a set of language-specific processes and general cognitive skills determines how the bilingual responds to language tasks.

Green's IC hypothesis predicts that language switching may take time because it involves a change in language schema for a given task and because of the time it takes to overcome the inhibition of the previously activated language. IC predicts that there will be such costs when switching among language tasks (i.e., translation and naming) as well as within specific tasks (i.e., language reception and production). The specific task investigated in this paper is regulatory processing in a lexical decision task for which there are empirical results from a study conducted by von Studnitz and Green [1997, Experiment 1]. In this study, German-English bilinguals are asked to decide whether or not a presented letter string (may be a word or non-word) was a word in L1 or in L2 using an alternating runs paradigm (i.e., there is predictable switching between languages). In the study, the color of the background on which the word was presented served as an external cue informing participants of the required language for decision. Figure 7.2 illustrates the relationship between this cue and two lexical decision schemas inhibiting one another, and the lexico-semantic system.

The SAS establishes the schemas which map an output of the lexico-semantic system (e.g., presence of an L2 tag) to a response (e.g., press left key if L2 word). The control mechanism is driven bottom-up and once established, the SAS monitors it to ensure desired performance. In the case of a new switch trial, a new schema is triggered by the external cue and suppresses the previously active schema. Moreover, a new word in a different language has to overcome the inhibition on its language tags from the previous trial. Thus, IC proposes two areas of inhibition: (1) schema-level inhibition and (2) tag inhibition in the lexico-semantic system. IC predicts that inhibiting a previously active schema and overcoming the inhibition of a previously active language will take time, manifesting as a switch cost.

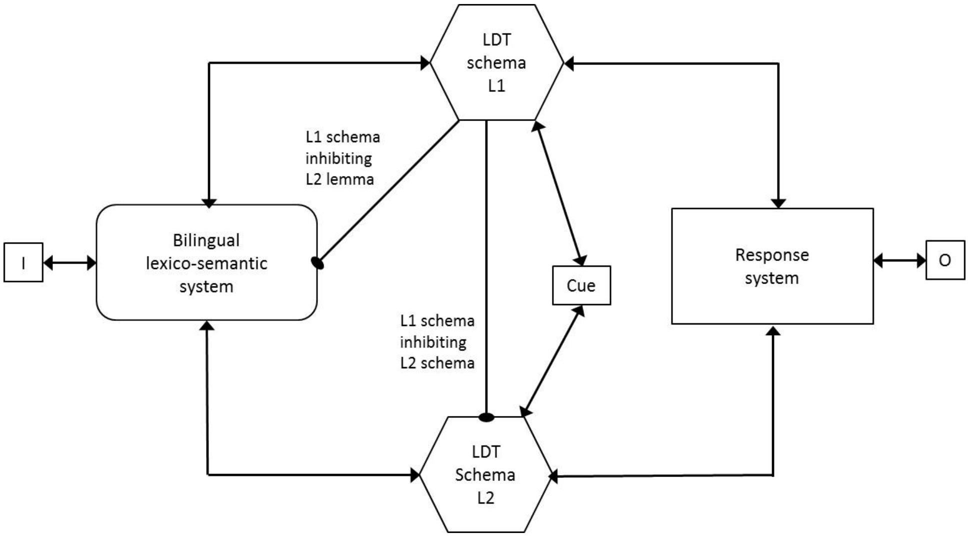

LDT schema L1
L1 schema inhibiting L2 lemma
I
Bilingual lexico-semantic system
Cue
Response system
O
L1 schema inhibiting L2 schema
LDT Schema L2

Figure 7.2: Regulatory processing in an LD task with language switching [Green, 1998]. The L1 task schema is suppressing the L2 task schema and inhibiting the L2 lemmas in the lexico-semantic system.

7.2.1.1 Empirically Testing the IC Theory

In the experiment [von Studnitz and Green, 1997, Experiment 1], language switches occurred on alternating trials (EEGGEEGGEE,etc.), indicated by a change in the color background which was counterbalanced across participants. Two types of stimuli were used: word and non-word, although both the IC and the computational model used only word stimuli. Each experimental block was preceded by a single filler trial which served to provide a clear designation for the experimental trial and in the case of the computational model, prime the lexico-semantic system and the task schemas to a “resting” state (i.e., activate the control mechanisms associated with either L1 or L2). Two sets of words were constructed with a total of 160 words in each: 80 words were English and 80 words were German in each set. In each case, half the words were high-frequency and half were of low-frequency. The words were matched for syllable length and letter length across the two languages. Words were orthographically possible in either language. Neither cognates (i.e., words that look the same and have the same meaning) nor interlingual homographs, "false friends" (i.e., words that look the same but have different meanings) were included. The experimental procedure is shown in Figure 7.3.

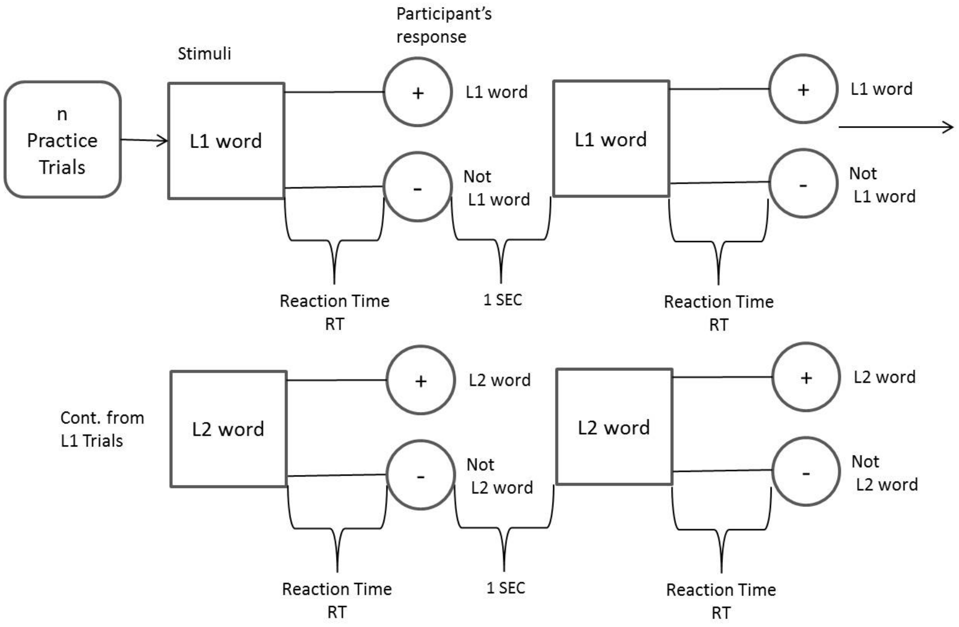


Figure 7.3: Experiment 1 procedure [von Studnitz and Green, 1997]. After n practice trials, participants are presented with a letter string and asked to decide whether it is a string in either L1 or L2. Language switches occurred on alternating trials indicated by a change in color background.

The experiment found an average switch cost of 118 ms for high-frequency and low-frequency English and German words and that participants were also 63 ms faster responding to German words compared to English words. The results from the experiment are summarized in Table 7.1.

7.2.2 IC Model Development

The purpose of the computational model is to verify the inhibitory mechanism proposed by Green [1998] and illustrated in Figure 7.2. Specifically, the goal is to verify the model's prediction of a cost when a bilingual switches from a required response in German to an English response for a lexical decision task, as well as from English

Table 7.1: Experiment 1 results [von Studnitz and Green, 1997]

Word Type	Switch Mean RT	Non-switch Mean RT	Cost
L1 German	805 ms	705 ms	100 ms
L2 English	887 ms	752 ms	135 ms
Mean	846 ms	728 ms	118 ms
Avg. cost $switch - nonswitch = 118\ ms$			
L1 RT advantage: 63 ms			

to German.

The design of the computational model is based on Green's IC hypothesis for regulatory processing in a lexical decision task as shown in Figure 7.2. An interactive activation and competition (IAC) connectionist model was built using a neural network simulation tool, NNSIM. One unit per processing pool was allocated since the likely distributed representations in the brain of the regulatory processes were not specified by the IC model and were not the focus of this study. The architecture of the model is shown in Figure 7.4. The model has three layers of units: an input layer of word representations that are connected to a hidden layer of lemma representations which are in turn connected to an output layer representing the lexical decision task schema. In addition, there are two input units representing the target language response cues used in the experiment (i.e., the background color on which the word is presented) each connected to its respective LDT schema. One instance of this three-tiered structure is provided for L1 and another for L2; the two are connected by inhibitory connections between the L1/L2 schemas, L1/L2 words, and L1 lemmas/L2 schemas, as supposed by the IC hypothesis. In Figure 7.2, a single cue is seen as connecting to both the LDT L1 Schema and LDT L2 schema, but it has been implemented as two separate color units to more accurately reflect the experimental procedure.

The entirety of the L1/L2 in the bilingual lexico-semantic system is not modeled here. Since the experimental stimuli presented across the trials were an average of high and low frequency English and German words, each L1 or L2 word

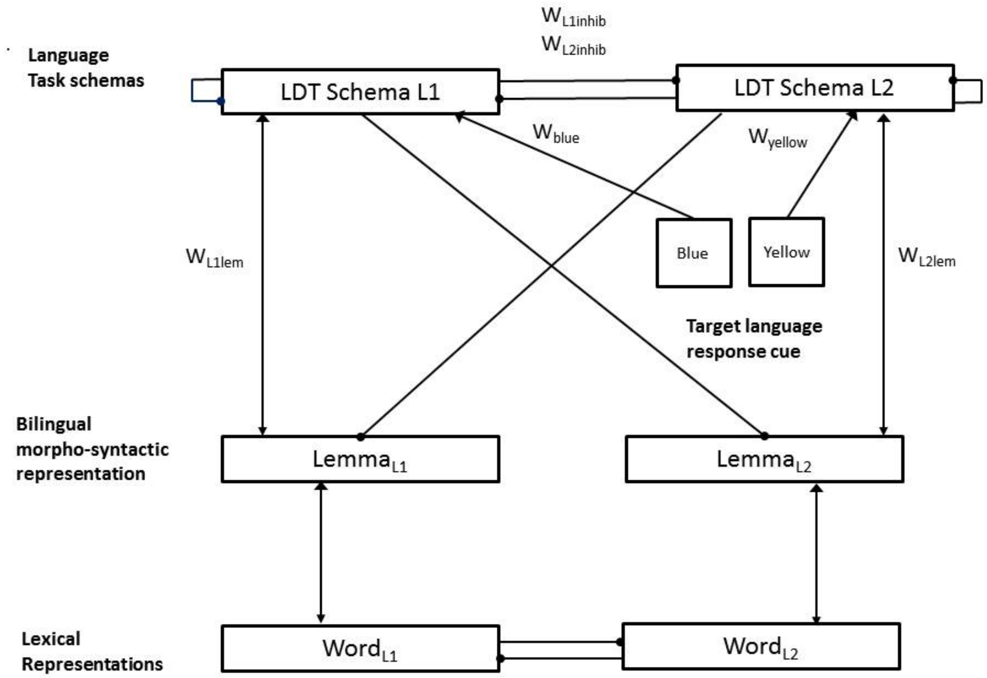


Figure 7.4: Model design. Stimuli are presented at the word level and activation in bottom up with language "tag" associated with words at the lemma level. The cue activates a task schema indicating whether an LD response should be in L1 or L2 The key IC feature are the inhibitory links between the L1/L2 LD schemas and between the L1(or L2) schemas and L2(or L1) lemmas.

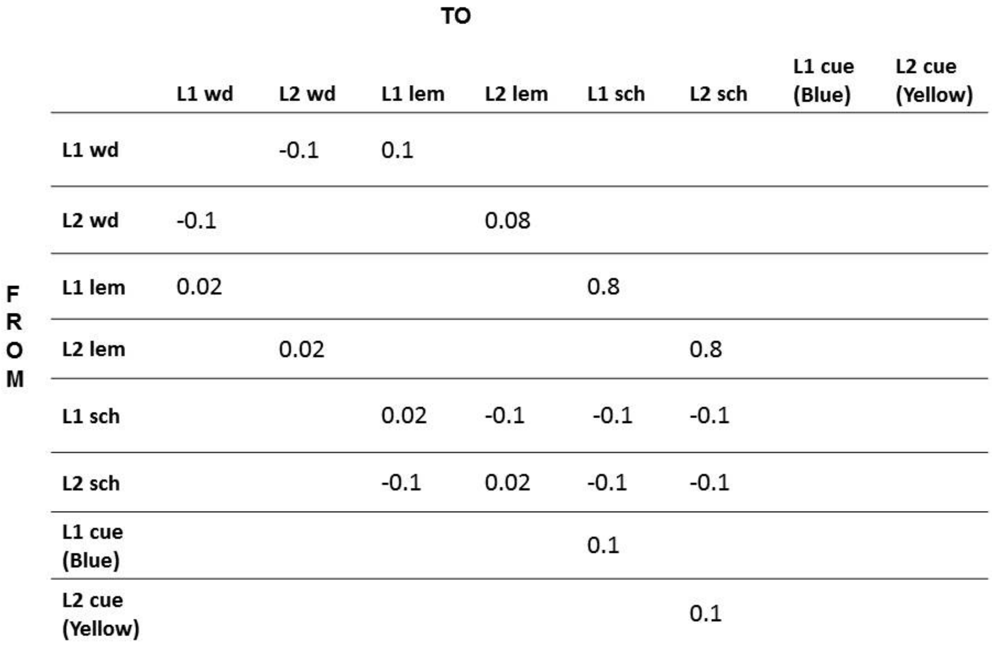
represents a sample word of average frequency from the trials. The lemma units are the morpho-syntactic representation of the word where Green [Green, 1998] posits the language tag is located. As with the word units, only the L1 and L2 lemmas connected to their corresponding L1 and L2 words are modeled. The L1 LDT schema and L2 LDT schema exist outside the bilingual lexico-semantic system and are the units that are monitored for their activation level.

The experiment measured a participant's reaction time, i.e., from when a word was presented on the computer screen to when the participants press the "+" or "-" key. It is apparent that the reaction time (RT) consists of two components: the time it takes to activate the schema plus the time it takes for the participant to move his or her arm and press the key. For the purpose of the output data mapping, only the schema activation time is of interest and the remainder of the reaction time is treated as a constant. Thus in the model only the number of cycles from the resting level of the schema until it settles at its activation level is measured.

Only the L1 structure was modeled to start. All the weights of the top-down connections were set identically. Three weight groups for the bottom-up excitatory connections were identified: (1) cue unit to LDT Schema, (2) word to lemma, and (3) lemma to LDT Schema. An input was applied to the L1 word unit and to the L1 cue unit and the bottom-up connection weights were adjusted until the L1 LDT schema was strongly activated, i.e., 0.800. This took place at 30 cycles. An identical L2 model was then constructed and the two structures were connected using the inhibitory connections hypothesized by Green. All inhibitory connection weights were set to -0.1. Even without further adjustment of the weights, we noticed a switching cost during a switch trial, but the cost was somewhat greater than what the empirical data suggested. However, by adjusting only the top-down connection weights uniformly, we were able to get a good fit with the empirical results for the language switch cost. In this iteration of the model, the network settles with the L1 schema activation at 0.793.

The symmetrical L1/L2 model however does not account for the L1 advantage observed in the experiment: participants in the study were 63 ms faster in responding to German (L1) words than to English (L2) words. Reasoning that the language effect was located in the lexico-semantic system rather than in the task schema, the weight of the connection between the L2 word and the L2 lemma was adjusted, producing the desired language effect and a good fit to the experimental data. With this change, the network settles with the L2 schema activation at 0.779.

We iterated through process of adjusting the top-down connection weights uniformly as a group and the connection from the L2 word to the L2 lemma until the summed square errors for the switching cost and L1 advantage of the model and experiment were minimized. This resulted in weights of 0.02 and 0.08 respectively; all the connection weights are given in Figure 7.5.



TO

FROM

	L1 wd	L2 wd	L1 lem	L2 lem	L1 sch	L2 sch	L1 cue (Blue)	L2 cue (Yellow)
L1 wd		-0.1	0.1					
L2 wd	-0.1			0.08				
L1 lem	0.02				0.8			
L2 lem		0.02				0.8		
L1 sch			0.02	-0.1	-0.1	-0.1		
L2 sch			-0.1	0.02	-0.1	-0.1		
L1 cue (Blue)					0.1			
L2 cue (Yellow)						0.1		

Figure 7.5: The connection weights depict the values found during model fitting.

7.2.3 Model Results

Table 7.2: Computational model results

	Switch	Non-switch	
Word Type	Mean RT	Mean RT	Cost
L1 German	830 ms	710 ms	120 ms
L2 English	910 ms	770 ms	140 ms
Mean	870 ms	740 ms	130 ms
Avg. cost $switch - nonswitch = 130$			
L1 RT advantage: 70 ms			

The experiment's reaction times (RT) needed to be mapped onto network update cycles in order to be able to simulate the temporal sequence of reading words and the sequence of internal cognitive processes during the activation of the task schema. This mapping was achieved by dividing the response times by 10 and rounding to the nearest whole number, thus one cycle = 10 ms. Using the mappings from the experiment's reaction times onto update cycles, a sequence of 10 trials was run as shown in Table 7.3 corresponding to the experimental procedure as shown in Figure 7.3. There are four types of trials: L1:initialize, L1:non-switch, L1/L2:switch, L2:non-switch, L2/L1:switch. L1:initialize represents a practice trial and it allows the network to settle at its L1 Task schema activation level. Although it is numbered as a trial in Table 7.3, "blank" is the inter-trial pause. We alternately apply input to the L1 word and Blue cue or to the L2 word and Yellow cue according to whether we want a switch or non-switch trial and then cycle through the network until the corresponding L1 or L2 schema is activated, recording the results.

In Table 7.3, the number of cycles (i.e., no. cycles) given for the non-switch and switch trials for both L1 and L2 corresponds approximately to the time it takes for the participant to ready a schema for making a lexical decision in the target language indicated by the cue, i.e., $LDTsch_t = RT_{avg} - k$, where k (i.e., Physical RT) is the time it takes for the response system to initiate the action to press the "+" key, and RT_{avg} is the average response time as measured in the experiment. Removing the inputs to both the lexical node and the cue for a period of 100 network cycles is the functional equivalent of the experiment's 1 second pause between trials. During this pause, we want the activated schema, lemma, and word to decay to represent the lower activation of the mental lexical, syntactic, and executive task control processes likely once the stimulus is removed. The resting activation levels of the word, lemma, and schema units represent either the base level from which we wish to return to activation if the next stimulus presented is from the same language as the previous, or the level which will be inhibitory to the rising activation of word, lemma, and schema from the new target language for a language switch trial.

Comparing the results of the von Studnitz and Green [1997, Experiment 1] study, Table 7.1, with the results of the computational model, Table 7.2, shows a good fit with the significant effect of cost switching as predicted by the IC hypothesis and suggested by the results from the empirical study (i.e., 130 ms for the model, 118 ms for the experiment). The model also exhibits the experiment's asymmetrical language RTs where participants responded faster to German words compared to

Table 7.3: Computational Model Trial Runs. Ten trials were conducted, the first being practice. The number of cycles is cumulative and the RT is computed from: $RT = (EventDuration(Trial_{N,a}) + EventDuration(Trial_{N,b}))$

Trial	Type	No. Cycles	Event Duration	RT (ms)
1	L1 initialize	32	32	
2	blank	132	100	
3a	L1 non switch	160	28	
3b	Physical RT	203	43	710
4	blank	303	100	
5a	L1/L2 switch	351	48	
5b	Physical RT	394	43	910
6	blank	494	100	
7a	L2 non-switch	528	34	
7b	Physical RT	571	43	770
8	blank	671	100	
9a	L2/L1 switch	711	40	
9b	Physical RT	754	43	
10	blank	854	100	830

Trial 1 is a “practice trial”
One cycle = 10 ms

English words (i.e., 70 ms for the model, 63 ms for the experiment).

This suggests that we were able to fit the model to the empirical data the best we could to achieve a “proof-of-concept” validating the IC hypothesis. Our results suggest that the inhibitory control mechanism is a possible explanation for the switch costs and asymmetric language RTs seen in the experiment, but the model does not yet provide strong evidence that it is likely the case.

7.2.3.1 Discussion

The inhibitory control model is important because it provides a theory for how bilinguals can perform different tasks given different language inputs. In addition, IC explains various effects observed in empirical studies such as, switch costs and unwanted language interference. IC has been an influence on other theories of bilingual word recognition and Dijkstra and Van Heuven [2002] incorporate aspects of Green’s theory in their BIA+ model. However, IC has only been specified descriptively at a functional level, thus lacking the advantages of a computational model. Their disadvantage is that, unlike computational models, functional models rely purely on behavioral experiments which can only superficially explore cognitive processes, and are not easily generalized, thus limiting their predictive ability.

Our computational model provides a framework for validating the inhibitory control model, and at least captures an important aspect of bilingual word recognition: the regulatory processing mechanism. After additional exploration of the model parameters it can be further developed and generalized to test what IC predicts in other tasks such as language switching in production. One weakness in many computer assisted language learning tools is their ability to train language learners to actually speak a new language. A computational model of regulatory processing in language production would add to our understanding of what inhibits a language learner from producing utterances in the new language.

Although the model provides a proof-of-concept that the IC regulatory mechanism described by Green [1998] is a possible explanation for the switching cost seen empirically, more evidence could be provided if the model incorporated recognition of non-words which were utilized in the empirical study [von Studnitz and Green, 1997, Experiment 1]. The inclusion of non-words makes a lexical decision task more meaningful and also provides a means for testing whether or not the nature of the non-word affected reaction time (e.g., English non-words possible in English only or in both German and English). Showing that participants are affected by the status of a non-word would provide further evidence against the input-switch hypothesis as non-words provide no route to the lexicon and therefore cannot use it to decode a response. Further, von Studnitz and Green [1997, Experiment 2] conducted an additional experiment using a generalized lexical decision task. In this experiment, participants needed to decide whether or not a letter string was a word in either language; the empirical study found a small but significant switching cost. Green underspecifies the inhibitory mechanism for a generalized LD task [Green, 1998, 74] and this would be an extension to the IC and the computational model. Neither the IC nor the computational model account for frequency effects or cross-language effects demonstrated in other studies of bilingualism and cognition, e.g.,, [Van Heuven et al., 1998]. Subsequent models of bilingual word recognition such as BIA+ [Dijkstra and Van Heuven, 2002] incorporate features of Green's IC hypothesis and include a model of the lexico-semantic system, which is not specified by Green. BIA+ also accounts for more of the empirical results observed in studies of bilingualism (e.g. orthographic neighborhood effects, cross-linguistic effects, non-linguistic context effects, stimulus-response binding). Further research in computation modelling of bilingual cognitive processes may well be better served investigating a more general architecture such as BIA+.

The IC hypothesis also predicts a cost in switching between languages in certain word production tasks such as numeral naming [Green, 1998]. Such tasks involve different language schemas and in order to produce speech, the activation of a new language schema would need to exceed the activation of the current language schema. However, this mechanism for doing so is not fully specified. Models accounting for speech production have been based on models by Levelt and Meyer [1999] and Dijkstra and Van Heuven [2002] discuss generalizing BIA+ to bilingual word production.

Beyond BIA+ there are novel approaches that provide a more dynamic view of the lexicon than the traditional connectionist network and combine localist and distributed properties of processing. One such model is the self-organizing model of bilingual processing, SOMBIP, [Li and Farkas, 2002]. It consists of two interconnected self-organizing neural networks, along with a recurrent neural network that computes lexical co-occurrence constraints. SOMBIP captures both bilingual production and comprehension and can account for patterns in the bilingual lexicon without the use of language nodes or language tags. It attempts to answer the question of where the information comes from that allows the bilingual to separate their two languages. A potentially interesting research direction is to examine Grosjean [1997] account of code switching, Levelt's speech production architecture, Dijkstra and van Heuven's BIA+, and Li and Farkas' SOMBIP create a computational model that can account the inhibitory control mechanism involved in code switching.

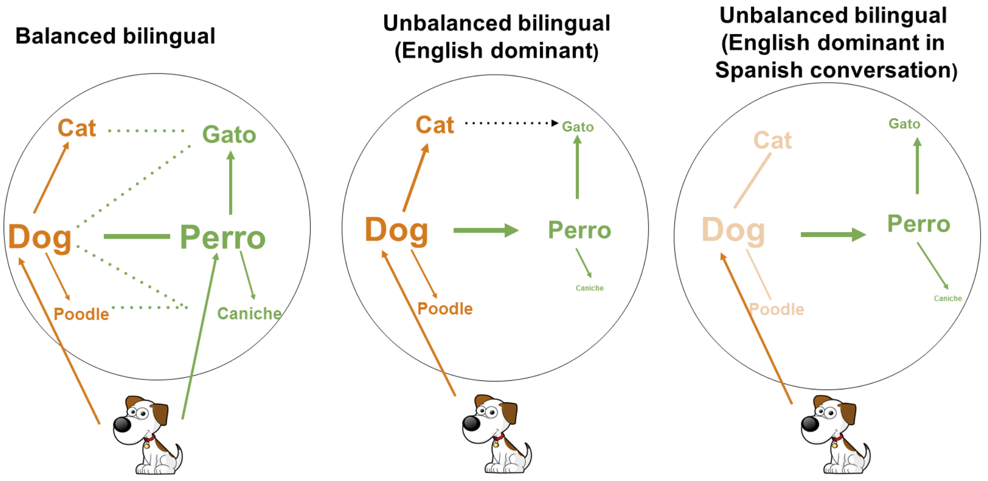


Figure 7.6: Bilingual Memory Theory. Baseline activations of lemmas is based on lexical frequency. Word co-activations are determined based on distance in semantic space. The activation strength of the translation equivalents is determined by the speaker's proficiency level.

7.3 Bilingual Memory

The inhibitory model of bilingual control uses task context to switch between the language lexicons (see Figure 7.4). In the investigation discussed, the context was a lexical decision task. More generally, the context can provide cues as to which language is to be used. In this section, we discuss our hypothesis of bilingual memory and show how access is controlled based on the conversational context and speaker proficiency levels. There are three parts to this hypothesis which we desire to test empirically by building a computational model: (i) bilingual conceptual memory, (ii) word selection, and (iii) language control. Respectively, these mean: (i) how the model represents a word other words that are conceptually related to it, in both the primary and secondary language, (ii) how a word is "chosen" from memory according to the language task, e.g., during speech production or reading, and (iii) how the word in the intended language is chosen to be spoken or recognized rather than its equivalent in the unintended language.

Our ultimate goal is to create a computational model of bilingual memory in which the language control processes adapt to the conversational context and change as the second language is acquired. We recognize this to be an ambitious, long-term project and decided that a good approach would be to break down the implementation into two or more sub-projects, each with its own set of goals. Since many issues which emerge in the study of bilingual processing involve the lexicon, we decided to begin our investigation here. We designed a simple computational model of the bilingual lexicon as a semantic memory network [Collins and Loftus, 1975] in which we can demonstrate how activation spreads from a concept to its lexical representations in a bilingual's first language (L1) and their second language (L2) and then to semantically related words in both languages. Furthermore, we restricted the semantic network to reflect those relationships that would likely be learned by a balanced English/Spanish bilingual.

In this section, we discuss the work we performed to build the model, review the research that led to the design, explain how the input data was mapped from the human experimental domain to the model, and discuss how we chose to represent the bilingual lexicon. Finally, we will explore how we parameterized the semantic relationships among the words in the lexicon using a large database of English and Spanish word embeddings, along with other information, e.g., word frequencies.

7.3.1 Bilingual Memory Hypothesis

Despite numerous, sometimes contradictory research studies, we remain far from a single psycholinguistic theory explaining how languages influence each other. For example, the empirical data these models draw on for testing are usually gathered in an artificial (i.e., laboratory), rather than naturalistic setting and this can lead to apparently contradicting results. It is has been recently shown, for example, that the cost of switching languages observed in the laboratory vanishes when repeated in a naturalistic setting [Blanco-Elorrieta and Pylkkänen, 2017]. Given that a bilingual robot would be interacting with humans, the design of its bilingual language component would benefit from a psycholinguistic theory of bilingualism that incorporates language behavior observed "in the wild". This lead us to investigate approaches that draw upon naturally-occurring behavior, specifically how language proficiency

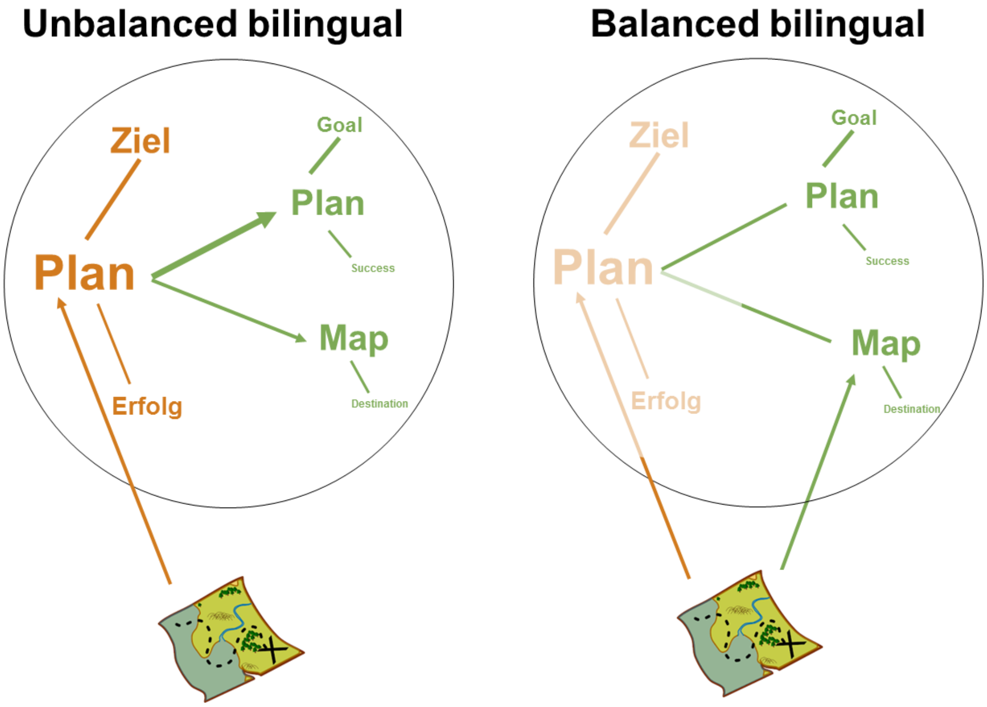


Figure 7.7: Bilingual Memory Development. Plan/Plan English and German. How do we explain that balanced bilinguals do not end up calling a map a "plan" in English but unbalanced bilinguals cannot discriminate between the two word senses? Unbalanced bilinguals, get to "Map" (in English) through the German word "Plan". Balanced bilinguals have a direct link between the concept of a map and both "Plan" (in German) and "Map" (in English), so they avoid this problem.

affects selection of the desired word and how co-activations of semantically-related words occurs in both the L1 and L2.

Figure 7.6 illustrates how the concept of a "dog" would activate its related English lexical representation which would then spread its activation to the three semantically related representations along with their Spanish translation equivalents. The figure shows this for three conditions: (i) balanced bilingual, (ii) Unbalanced bilingual, who is English-dominant, and (iii) unbalanced bilingual, who is English-dominant and engaging in a Spanish conversation. In Figure 7.7, we illustrate a developmental condition for German-English bilinguals. As bilinguals become more proficient and move towards becoming balanced, connections are formed directly from the concept to the related lexical form of the second language, a route which bypasses the related word form of the first language.

Figure 7.7 provides an explanation for why balanced bilinguals do not end up calling a map a “plan” in English, but unbalanced bilinguals do. Plan is a homograph in English and German. Homographs are words that share a similar word form but have *different* meanings in L1 and L2; they are often referred to as “false friends”. Unbalanced bilinguals, get to the English word ”map” through the German word “plan”. Consider that the German word plan has an activation level of 5. “Plan” in English has an activation level of 2 and “map” in English, which is the target word, has an activation of 3. The word “plan” received an unwanted boost in activation, influenced by the activation for the German “plan”, making it the most available candidate for selection, even though it is not the target. On the other hand, balanced bilinguals have developed a direct link between the concept of a map and both “plan” (in German) and “map” (in English), so this problem is avoided. The theory also assumes that top-down control process provides inhibition to the German lexicon, based on the task context of answering in English.

Our hypothesis is also informed by the experiment conducted by Blanco-Elorrieta and Pylkkänen [2017] which found no language switching cost when bilingual speech production occurs outside the laboratory in a “naturalistic” setting, using facial rather than artificial cues to signal the desired target language in a picture-naming experimental paradigm. A similar experiment was conducted for comprehension and no switching cost was found as well. Thus the language switching cost debate remains to be settled and a computational model that attempts to explain these somewhat contradictory accounts is one of our goals. In the following sections, we review the primary issues we had to tackle for the model design.

7.3.2 The Modified Hierarchical Model

Understanding how cross-linguistic differences operate at the conceptual representation level has been the focus of bilingual research over the past decade. In [Pavlenko, 2009], Pavelenko provides a comprehensive review of the models of the bilingual lexicon and proposes a new approach, the *Modified Hierarchical Model* (“MHM”). It

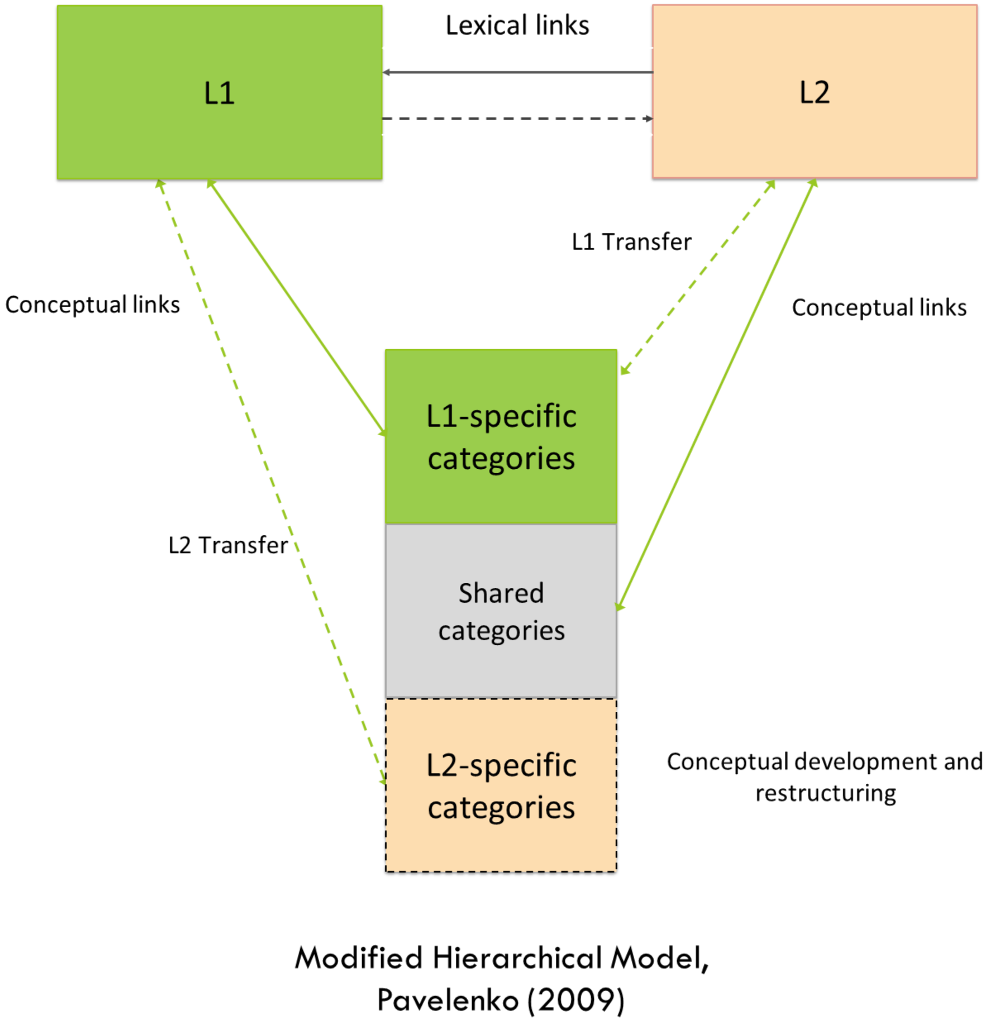


Figure 7.8: The Modified Hierarchical Model

differs from the major models Pavelnko reviews in three ways (see Figure 7.8). First, in the MHM, conceptual representations may be fully shared, partially overlapping, or full L1/L2 language specific. One implication is that only one language may have the necessary word forms and thus activating links to the other language may fail, producing disfluencies. In this case bilinguals may resort to code switching or lexical borrowing to continue the conversation.

The second characteristic of the MHM model is that of distinguishing between *semantic transfer* and *conceptual transfer*. Pavlenko gives the example of a Finnish speaker of English who mistakenly uses the word *language* for *tounge* as in

"He bit himself in the language". Both Finnish and English differentiate the two concepts that *tounge* can be used for so the speaker has made a semantic transfer error, by linking to the higher-frequency English word *language*; it occurred at the level of mapping words to concepts, not involving the structure of conceptual categories. Figure 7.7 gives another example of a semantic transfer error, where the L1 German word *plan* is linked to the high-frequency L2 English word *plan* rather than *map*. On the other hand, when an English speaker of Russian uses the word *chashka* for a paper drinking cup, they have made a *conceptual* error as *chashka*, while similar to a cup, does not include the category of paper or plastic containers.

The third characteristic is that L2 learning is embedded in the model. It is seen as a gradual process, taking place in implicit memory (i.e., individuals may not be aware that they are acquiring language knowledge "in the wild" rather than by learning language rules as in explicit memory). MHM views L2 learning to be a gradual conceptual restructuring with the goal being to acquire target-like linguistic categories. This distinction differs from Kroll and Stewart's Revised Hierarchical Model ("RHM") [Kroll and Stewart, 1994] in that RHM assumes the goal of L2 vocabulary learning is to develop direct links between L2 words and concepts. Pavlenko's main argument for differentiating between implicit and explicit learning is to provide a model of second language vocabulary learning that emphasizes the structure of the conceptual representations rather than the interlingual connections as in other models like RMH. Pavlenko does not imply that other models of bilingual processing should include an account of cross-linguistic differences in linguistic categories. Rather, she argues that models of L2 vocabulary learning and bilingual lexicon models would benefit the most and that by focusing on these representations, a better understanding of vocabulary learning will emerge.

We can see the benefit of considering aspects of the MHM model in our computational model of bilingualism. It appears to be a more ecological sound model, accounting for speaker vocabulary errors not in explained by other models. As Pavlenko points out, it is a starting point for further empirical studies including a long-term longitudinal study. At this point, MHM is psycholinguistic model and operationalizing its distinct characteristics, e.g., organizing the conceptual store into three parts rather than having a single unified store, differentiating conceptual and semantic transfer, and gradual conceptual restructuring is a challenge. In the next section, we describe our preliminary attempt to build a computational model inspired by RHM and MHM.

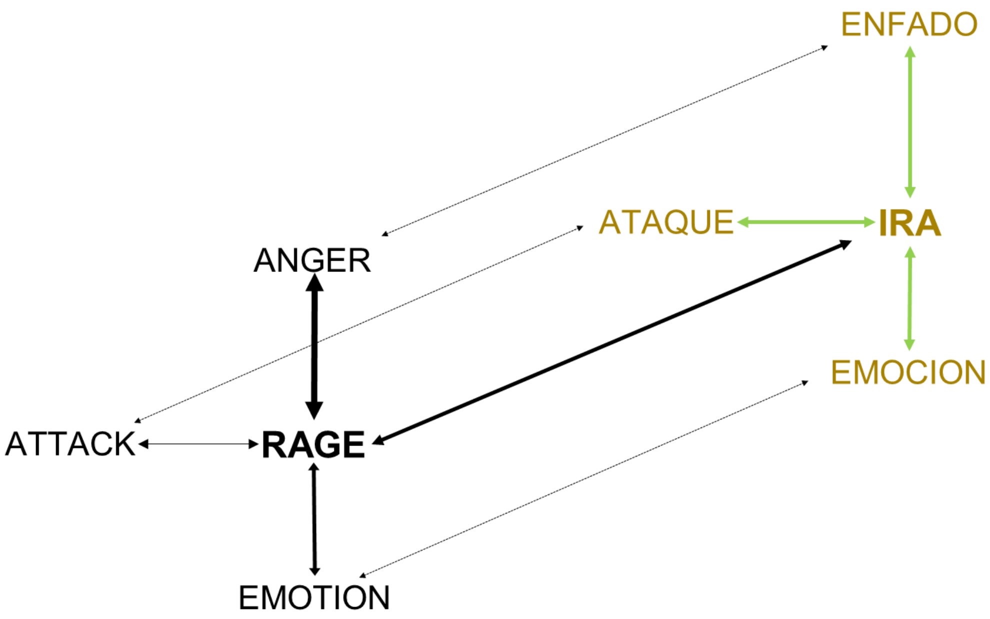


Figure 7.9: Spreading-activation theory applied to bilingual semantic memory processing.

7.3.3 Representing the Bilingual Lexicon

The first task was to select the computational modeling paradigm used to implement our theory. A computational model differs from a psycholinguistic model in that it is specified using a computer programming language and hence is able to “run” on a computer, simulating the cognitive processes as specified by the theory. We decided to use connectionist modeling, and in particular a class of designs known as Parallel Distributed Processing (PDP) as defined by McClelland and Rumelhart [1989]. The PDP design has historically been used to gain insight into the cognitive processes of the bilingual mind, and the BIA [Dijkstra and Van Heuven, 2002] is one such model. We used this paradigm to design the implementation of our hypothesis.

The challenge in building a computational model of bilingual lexicon theories such as RHM and MHM is how to learn the strengths of the links shown in Figure 7.8. The RHM theory has been implemented as a PDP model in which the weights were adjusted until the desired network dynamics were observed [Sadeghi et al., 2013, Laszlo and Plaut, 2012b]. Dijkstra et al. implemented a computational model for bilingual word recognition and word translation, Multilink, whose connection weights are also set by observing the model behavior and manually adjusting the weights until the desired performance is achieved [Dijkstra et al., 2019]. Rabovsky and McRae [2014b] trained the weights in their monolingual PDP model of word meaning using backpropagation. Their model extends the attactor model of conceptual processing desribed by Cree et al. [1999] which was trained to map word forms to human-generated features. Here, semantic memory is represented by 190 semantic feature production norms that were determined by human participants in a norming experiment. While this is a more ecologically valid approach than having the researchers set the representations, it is limited by the size of semantic feature production norms. Adding new concepts means adding new feature production norms and conducting new experiments. We will now describe our approach to representing the bilingual lexicon as embedded word vectors, which is not limited to an arbitrary feature set size and does not need to be normed.

The semantic memory model described in [Cree et al., 1999] is an *associationist* network. In this type of network, word meaning is represented by how often it occurs with another word; this relationship is captured using high-dimensional "semantic vectors". In such a representation, a single concept might have 200 to 300 dimensions (i.e., distinct values) each representing a numerical measure of a semantic feature of the word; note that it is not obvious what specific semantic properties these dimensions represent. The closeness of two words can be determined by measuring the geometric distance two such vectors are from each other in semantic space; computations such as cosine similarity or Euclidean distance are common measures. In our model, word vectors are drawn are from an associationist database such as the Hyperspace Analog to Language (HAL) [Burgess and Lund, 1997]. A major advantage of these lexical databases is that they are available for many languages other than English and that they are trained on naturalistic corpora such as Wikipedia or movie subtitles. There are deep learning computational methods, such as Word2Vec [Mikolov et al., 2013], that allow the researcher to build their own corpus of word vectors on, for example, a conversational bilingual corpus such as the Bangor Miami corpus [Deuchar et al., 2014].

Collins and Loftus [1975] theorize that semantic processing occurs in an associationist network through *spreading-activation*. According to this theory, a concept can be represented as a node in a network. The search in memory between concepts involves following a path in parallel along the connections from the node of each concept specified by the input words. The spread of activation constantly expands, first to all the nodes connected to the first node, then to all the nodes connected to each of these nodes, and so forth until some unspecified depth. Furthermore, the semantic network is organized along the lines of semantic similarity; concepts that are similar are linked together. Activation levels are affected by the strength of the connections. Collins suggests that spreading activation has a neurological basis and McClelland and Rumelhart's PDP models incorporate this concept.

Figure 7.9 shows a portion of a spreading-activation model of bilingual memory. Here, the L1 English word "rage" evokes three related words, "anger", "attack", and "emotion". In addition, "rage" evokes the L2 Spanish translation equivalent, "ira" which in turn evokes the related words "enfado", "ataque", and "emocion". Note that the L2 evoked words may be different than their L1 translation equivalents, although they are the same here. We made a simplifying assumption that the Spanish evocations are identical to their English translations; this likely not to be accurate.

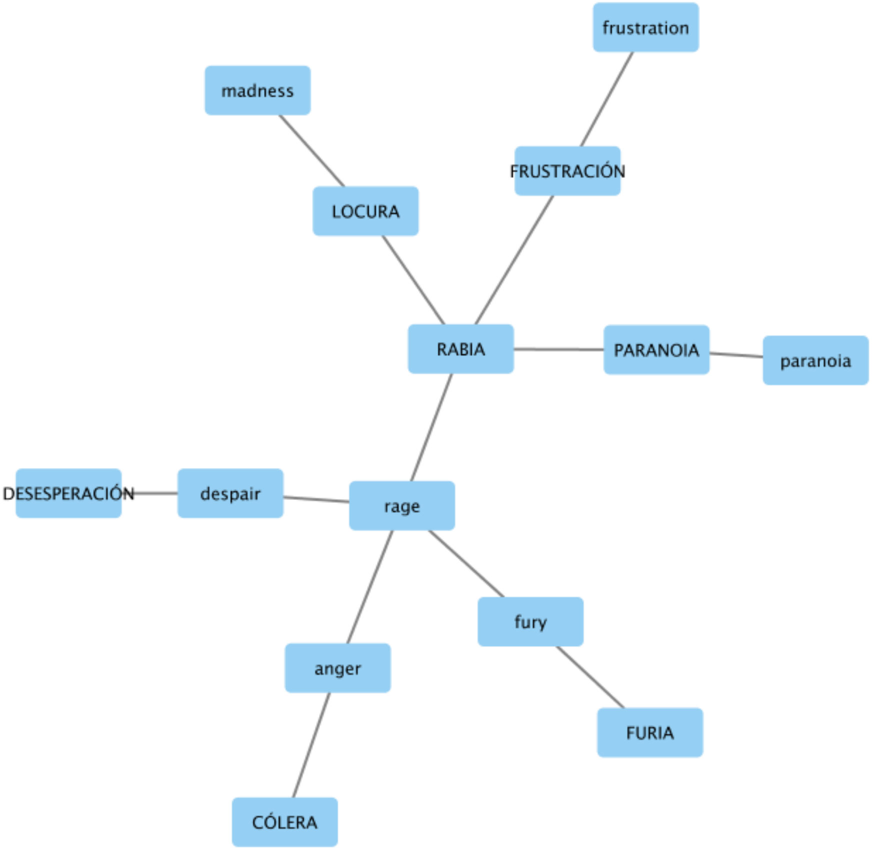


Figure 7.10: Bilingual lexicon generated for the L1 English word *rage*. Parameters to the model $b = 3, d = 1$ generate three evocations at the first level and only their translation equivalents. As can be seen, the L2 Spanish evocations are different than their L1 English counterparts.

7.3.4 Bilingual Memory Model Design

The computational model consists of a *bilingual lexicon* and a *bilingual memory* component. We represent the bilingual lexicon as a bidirectional graph whose edges connect related words; the degree of relatedness between two words is captured by the edge's connection strength (see Figure 7.10). A computer program generates the graph when given a list of English words and parameters b (breadth) and d (depth), the model will recursively generate a co-activation graph of the b words immediately evoked, and the b words each of those words invoke, to a depth d. At every depth it will find the Spanish translation equivalent for each English word and similarly recursively generate the b words they invoke to a depth of d. The strength of the connections between the evoked words is proportional to their distance in semantic space i.e., words that are more distantly evoked will have a lower connection strength. The computer program used the Microsoft Translator text API [Mic, 2018] to automatically translate between English and Spanish words. For the L1 English words, semantic distance is obtained from a database of learned vector space word representations trained using GloVe on 2014 Wikipedia [Pennington et al., 2014]. L2 Spanish words were selected from a database of learned vector space word representations trained using GloVe on Spanish Wikipedia [Etcheverry and Wonsever, 2016]. The computer program generated the bilingual lexicon as a graph adjacency list which the bilingual memory component uses to construct the processing component.

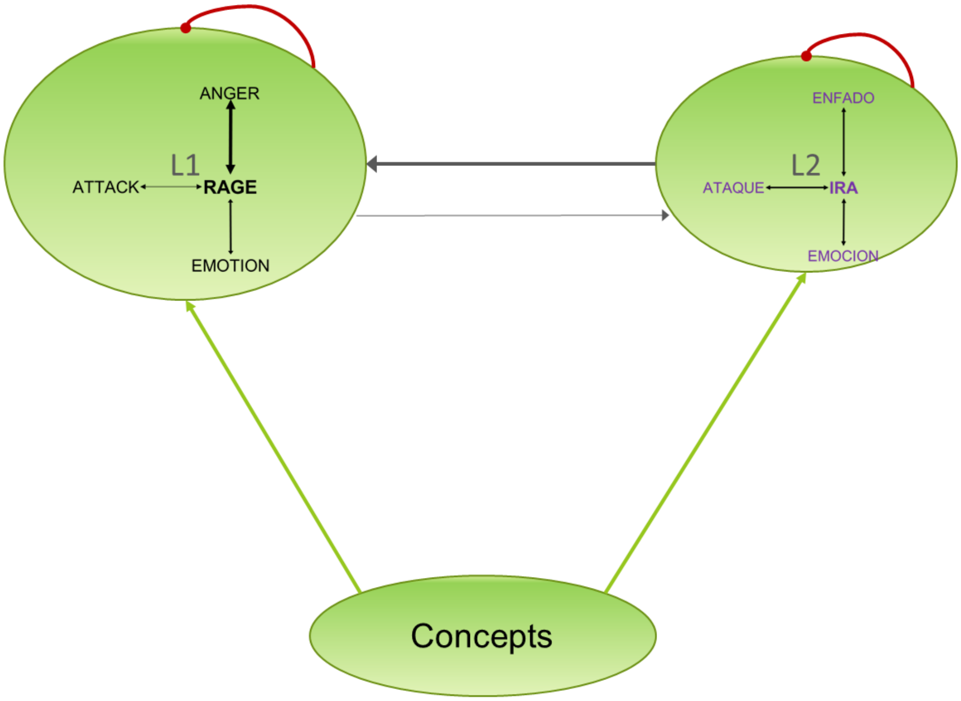


Figure 7.11: Computational model of bilingual memory processing.

The bilingual memory component implemented the bilingual lexicon processing (i.e., word selection). In its initial implementation, it is similar to the Revised

Hierarchical Model and the shared, distributed, asymmetrical models described in [Kroll and Stewart, 1994, Dong et al., 2005], although we would like to include the semantic feature categories of the Modified Hierarchical Model in future development. In these models, as a language is acquired, connections to the primary language representations are formed and strengthened through repeated association of the concept with its word form. Similarly, word forms in the second language become first associated with primary language word forms and over time, connections develop directly between the second language form and the concept. At present, our model implementation does not simulate this effect of language acquisition It is responsible for correctly selecting the L1 or L2 word from the bilingual lexicon.

The model implementation uses the Parallel Distributed Processing (PDP) framework McClelland and Rummelhart described in [McClelland and Rumelhart, 1989]. In the PDP framework, words are activated through spreading activation as in Collins' model of semantic memory networks. As shown in Figure 7.11, concepts are connected to their lexical representations in both the Spanish and English lexicons and these are in turn connected to their evoked words and their translation equivalents. Information flows unidirectionally from concepts to their lexical representations which then become "excited" or activated. These excitatory connections are bi-directional and are mathematically represented as floating point numbers. Activation next spreads to the evoked words and to all the translation equivalents, all of which in turn spread their activation back in the reverse direction through the connections. This provides supporting "evidence" to the original word and its evocations (but not to the concept). Note that evoked words do not excite other words in the same set.

Alternatively, connections in the lexicon can be inhibitory and these are represented as negative floating point numbers; we have set such unidirectional connections to be -1.0. In the example using the concept "rage", once all its evocations and translation equivalents receive activation and start spreading it back through the connections, the evoked words in the ovals start inhibiting others in the set (e.g., "anger" inhibits "attack" and "emotion"). The purpose of these connections is to suppress activation of a word by other, competing sources of information. Examples might be the need to produce another word in the flow of conversation; or it could be the need for an unbalanced bilingual to suppress the stronger language when speaking in the weaker language.

The component's connections are initialized from the adjacency list generated by the bilingual lexicon component. This list contains the information about how the nodes are to be connected, the strength of the connections, and the starting, "resting" activation of the the nodes. As described by McClelland and Rumelhart [1981a], word frequencies are used to set the resting activation values. English word frequencies were obtained from The English Lexicon Project [Balota et al., 2007] and Spanish word frequencies from Mark Davies, Brigham Young University [Davies, 2018].

7.3.5 Model Processing

The processing dynamics in our implementation are based on a class of models known as interactive activation and competition (IAC) models which also belong to the PDP design class mentioned previously. The theory of these models suggests information flowing "top-down" from higher cognitive processing levels combines with information flowing "bottom-up" from lower processing levels to provide a set of constraints out of which aspects of language processing, such as lexical selection, arise [McClelland and Rumelhart, 1981a]. Information flows through a spreading activation process so that information at one level spreads to neighboring levels above and below. The notion of higher and lower cognitive processing levels is common in the literature; at the very highest level might be contextual influences such as pragmatics, interlocutor language, or the register. At the lowest level might be visual or phonetic features produced by the sensory input process.

Our model consists of sets of units, roughly corresponding to neurons, divided into three pools: (1) concepts, (2) English lexicon, and (3) Spanish lexicon; as the model develops there will be additional pools. The units are connected as described in the previous section.The activations of the units evolve gradually over time in a continuous fashion. However, when simulating this model computationally this mathematical ideal is approximated by breaking time into a sequence of discrete steps called cycles. At the start of every cycle, the activation value of every unit is the value that was computed at the end of the preceding cycle. Here is how the network computes the values:

1. Compute the input values to each unit

2. Compute the activation of the units

This two-step procedure ensures that nothing is done with the new activation of any of the units until all have been updated (i.e., the update is synchronous).

7.3.6 Model Input and Output

The model has a simple character-based menu that allows the user to enter data, control processing, and analyze results. Prior to running the model, the word frequencies need to be loaded via a menu command. In the future, this functionality will be expanded to allow an entirely new lexicon, and their frequencies, to be loaded rather than using the default set of 28 words. Concepts are entered into the model by typing the name of one of seven concepts. Model processing is controlled by repeatedly pressing the cycle key; we envision presenting a new concept and cycling as driven programmatically by a script.

The user can view the model state through one of several display commands which show the how the activation values of a word or a set of words evolve over the time period represented by the number of cycles processed thus far. In this way, the gradually spreading activation from concept through the words in the lexicon can be visualized. In addition, the model attempts to be as informative as possible, communicating its progress as well as informing the user of any errors it encounters. The goal is for the program to always handle error conditions gracefully and without terminating unexpectedly.

7.3.7 Discussion

As a first step toward a naturalistic computational model of the bilingual lexicon, we constructed a preliminary computational model to show how access in controlled based on conversational context and proficiency levels. The model demonstrates how a concept activates a word and, through spreading activation, activates evoked words. Words and their evocations are modeled as a semantic network of words and their evocations. Similarly, the representation of a concept completely shared between the languages is overly simplified, and likely captures the lack of differentiation that one might observe in early stages of language acquisition. Concepts shared between languages could be captured if the semantic networks was built from highly proficient bilinguals in which the L1 and L2 semantic categories are already differentiated. At this point, it is not a model of second language acquisition and cannot be trained to become more proficient. There would be separate models for unbalanced and balanced bilinguals with different parameters. The model has not yet been connected to the top-down inhibition IC mechanism described earlier. This mechanism would be recruited in a situation where an unbalanced bilingual would be speaking in their L2, for example. While not fully functional, the bilingual memory computational model provides the scaffolding needed for further research. For example, we implemented a way for the researcher to interact with the model, entering input, controlling its operation, and viewing the processing results; there are also a few basic analytic tools to help interpret the model's processing.

The initial model implementation is a way to build scaffolding upon which more aspects of the theory can be implemented. It provides an architecture for interacting with the model which can expand and be modified as needed. The implementation gave us an opportunity to test some aspects of the spreading activation theory and identify areas that need further investigation. Among the improvements that could make the model more viable in a bilingual robot include modifying the model to be more sensitive to the interactional context; one such a theory is described in [Green and Abutalebi, 2013]. Creating a bilingual language model which attempts to predict the next word and the language which it may be produced in, given the context of prior words in the conversation and any top-down cues would be necessary if the model is to emulate code switching. Code switching between languages is a common occurrence between two balanced bilinguals which may give the bilingual robot a more natural conversational partner. As noted, at present, the model does not incorporate language acquisition. The model's connections are initialized by the bilingual lexicon component which at present uses English and Spanish word vectors trained on Wikipedia corpora. Using actual conversations from bilingual speakers at different stages of second language acquisition to train the word vectors could provide a way to bootstrap the lexicon with node links and connection strength that correlate with the different developmental stages. The would allow the bilingual robot to emulate a speaker at different proficiency levels. Incorporating these features into the model would require further investigation and validation.

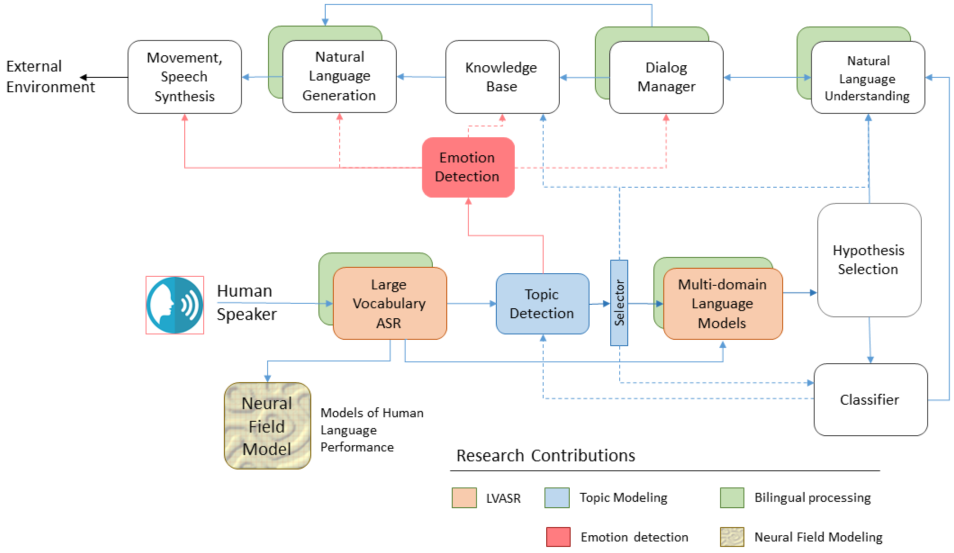


Figure 7.12: Spoken Dialog Framework: The SDS consists of the Large Vocabulary Automatic Speech Recognizer (LVASR) which uses Topic Detection to automatically select from among the Multi-domain Language Models. Transcribed speech is passed to the Emotion Detection component which forwards its prediction to the components as shown. Bilingual processing is implemented in the indicated green components. Models of language performance can be implemented using a Neural Field Model, which receives input from the LVASR. Dotted-line connections indicate proposed functionality that has not yet been implemented.

7.4 Implementing a Bilingual Robot

Figure 7.12 shows the components of the multi-level model framework. The Emotion Detection component is described in Chapters 4 and 5; the remaining components are described in Chapter 6. In the figure, modifications to the Natural Language Generation (NLG), Dialog Manager (DM), Natural Language Understanding (NLU), Large Vocabulary ASR (LVASR), and Mutil-domain Language Models (MD-LM) components to incorporate bilingualism are shown, except as noted below, in green. the following high-level description of such modifications will demonstrate the viability of building bilingual robot.

In order for the robot to understand the languages of the bilingual speaker, the acoustic model ASR component will have to be modified to extract the acoustic features of multiple languages. There a several ways of doing so. For example, Goshal et al. describe using a deep neural network which maps the features to tri-phones in a given language then using this to iteratively train more languages. The DNN is initially trained on a "seed" language and then a Softmax layer is added to a new language and DNN is fine-tuned to this language. The Softmax layer is repeatedly replaced by one corresponding to a different language and the fine-tuning is done for each [Ghoshal et al., 2013]. After training, the DNN outputs are then used as the likelihood states of a Hidden Markov Model (HMM) and the authors measure the word error rate of the multilingual system compared with a monolingual for a given language. Depending on the language, results comparable the monolingual are reported.

As discussed in Section 7.1, the LVASR arrives at its hypothesis by incorporating the probability that the predicted word appears in the language context. This is the role of the language model and thus there needs to be a way to either have a single LM for all the languages the multi-lingual LVASR recognizes, or there there must be multiple LMs that can freely switch among languages, controlled by the beliefs about the world (e.g., bilingual interlocutor), the task demand (e.g., start language understanding in L2), or context (e.g., the topics in the utterance belong to a different language). Figure 7.12 shows the *Topic Detector* (in blue) which detects context in an utterance and can signal the *Selector* component to switch to another Language Model.

Similarly, there are multiple Natural Language Understanding components which can signal a reinterpretation of the utterance using another LM by sending a signal to the Detector. It is in the NLU component we propose incorporating the model of bilingual memory discussed in Section 7.3.4. This allows the NLU to incorporate activation of non-target words in its semantic interpretation and to update the Knowledge Base. The parser in the NLU might use this, for example, to prime the selection of the correct frame when code switching, or to make a fine categorical distinction and avoid conceptual errors (see Section 7.3.2)

In addition, there might be multiple Dialog Managers depending on whether the dialog is to be conducted in, for example, a single language, more than one language, or code-switching within a sentence. Finally, the agent may generate speech in any of the languages and may code-switch. Therefore we show multiple Natural Language Generation components for these situations, in green. For a further discussion of how language model switching works, and for a demonstration applied to improving semantic interpretation in a spoke dialog system, refer to Chapter 6.

7.5 Summary

In this chapter, we have proposed that in a world where, depending on the region, one-quarter to one-half of the population speaks more than one language, it is increasingly likely that social robots, assistive robots, and conversational agents will be equipped with a bilingual capability. We summarized research on extending the acoustic and language models to recognize multiple languages, mostly driven by the need to incorporate recognition of under-resourced languages. Providing a way to allow the agent to code-switch remains an area of active investigation. We observe that many of the multi-language designs described in the literature are not rooted in the psychological models of bilingualism, e.g., language selectivity, cross-language interference, language switching. Therefore, it is unknown whether these phenomenon which are observed in bilingual speakers are important to bilingual understanding. For example, whether or not activating words or larger structures in the non-target language primes code-switching is an interesting research question. Furthermore, precisely which information cues and what they contain required to prompt switching between languages or in code-switching, is understudied in the context of bilingual robots.

Thus, we conducted two investigations: a computational model of language control and, a model bilingual memory. From this, we demonstrated how to extend a Spoken Dialog System (SDS) to incorporate bilingualism. We validated Green's IC in the context of VonStudnitz and Green's LDT experiment and conclude that it is a viable method of providing top-down control for language switching. We incorporated this concept in our SDS framework, where it is implemented as the Topic Detector and Selector. While we have not yet validated our model of bilingual memory (we propose doing so though a primed lexical decision task), we demonstrated operation of the model as described in Section 7.3.4. Finally, in 7.4 we demonstrated the connections and functions that are necessary to implement bilingualism in a spoken dialog system.

Chapter 8

Modeling Human Language Processing with Neural Fields

Human cognitive performance is very resilient in its processing of spoken language, yet ASRs do not achieve human-level performance nor do they attempt to replicate human spoken language processing. We investigated a biologically-plausible account of speech perception that is based on dynamic field theory [Schöner and Kelso, 1988]. Our model is an implementation of neural fields, which are hypothesized to be the basis of processing in the neocortex [Amari, 1977]. In our model, speech signals are presented in real-time and cause a neural field to fall into a stable pattern. These patterns can be associated with speech category levels at different levels of cognitive processing. Neural fields can account for two notable characteristics of human speech perception: robustness to noise in the environment and listener ability to reliably identify the speech sound from different speakers.

In the first case, neural fields are robust to moderate amounts of noise, remaining in their settled equilibrium pattern. It takes a nontrivial input signal to perturb the field from its equilibrium to a new state. In the second case, the neural fields are not sensitive to the absolute acoustic input signal but to how the input signal changes through time. Thus, it is this change over time which acts as a normalizing function, allowing speech perception across different speakers.

In our first investigation 8.1, our neural field model successfully replicated the effect of immediate auditory repetition of monosyllabic words and fits it to a component of a well-studied mechanism for analyzing language processing, the event-related potential (ERP). This represents a new modeling approach to studying the neuro-cognitive processes, one that is based on the bottom-up interaction of real-time sensory information with higher-level categories of cognitive processing.

In a subsequent investigation, we described a two layer neural field model in which category perception arises from the incremental recognition of temporal patterns from sequences of inputs, accomplished by decoding the neural field. In an example application, we used these patterns to identify a set of words which share the word onset represented by the input sequence, consistent with the Marslen-Wilson COHORT model of word recognition. Similarly, we evaluate the extent to which information contained in the bottom-up sensory signal can be used to determine word boundaries. We suggest it is plausible that a neural field offers a naturalistic explanation of how perception arises in word processing.

8.1 Correlating with a Human Physiological Measure

Previous attempts at modeling the neuro-cognitive mechanisms underlying word processing have used connectionist approaches, but none has modeled spoken word architectures as the input is presented in real-time. Hence, such models rely on the ingenuity of the modeler to establish a mapping of real-time stimulus to the model's input which may not preserve processing that happens during each time step. We present a neural field model which successfully replicates the effect of immediate auditory repetition of monosyllabic words and fits it to a component of a well-studied mechanism for analyzing language processing, the event-related potential (ERP). This represents a new modeling approach to studying the neuro-cognitive processes, one that is based on the bottom-up interaction of real-time sensory information with higher-level categories of cognitive processing.

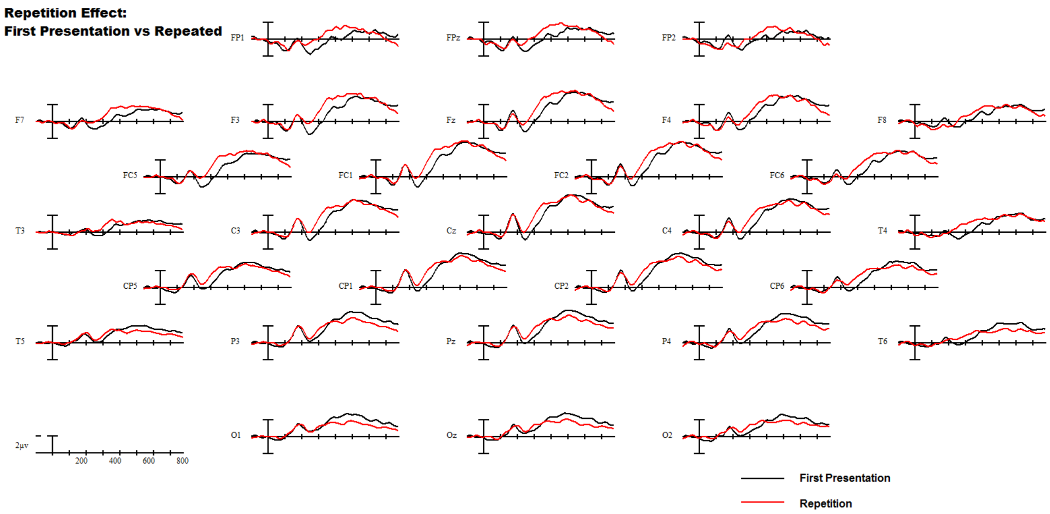


Figure 8.1: ERP repetition effects, seen in the difference between the first presentation (black line) or a word and the immediate repetition (red line) of that word

8.1.1 Introduction

By *spoken word perception*, we mean the cognitive processes that entail the sensory intake of an acoustic waveform until the words contained in it are identified. Some early connectionist models of speech perception processes were driven by research in generalized automatic speech recognition and have shown, for example, that a good deal of phonemic information is present in the auditory signal and can be extracted from the statistical generalization of the model. Among the best-known models of speech perception is TRACE [McClelland and Elman, 1986] which has modeled several lexical effects (e.g., phonemic restoration in a noisy environment) and the time-course of word recognition. TRACE has been criticized for its biologically unrealistic handling of time and the lack of a learning mechanism [Protopapas, 1999]. As a result, models were developed [Elman, 1990, Norris, 1995] which represent time through cyclical, "recurring" connections from one state to an earlier state in the network. One popular method by which learning is incorporated in these networks is through a gradient decent regression using backpropagation.

While these models can account for many aspects of how humans comprehend spoken and written words, none of these architectures model speech perception using real-time, human input. We present a neural field model with an efficient learning mechanism which dynamically responds to the spoken word process as it unfolds over time. A neural field sits in an equilibrium state waiting for a pattern it has tuned itself to detect, and this detection takes the form of a perturbation. Learning associates the equilibrium state of a field with its environment. Primary fields tune themselves to fall into systematic equilibrium states in response to combinations of sensory input. Deeper-processing, secondary neural fields are then enabled to tune themselves in response to their environments once primary fields have settled into predictable behaviors. With experience, the network forms representations as each neural field systematically responds to its environment through time.

8.1.1.1 Word Repetition Effects and ERPs

An event-related potential (ERP) is an electrical voltage associated with an event such as a stimulus or response. ERPs are believed to reflect the summation of post-synaptic potentials occurring in many thousands of neurons. The time course of ERPs in auditory processing can be traced starting from stimulus onset and continuing for approximately 800 ms. Our study focused on a particular ERP known as the P200 (P2) which occurs in the interval from 145 ms to 225 ms after stimulus onset and is classically associated with top-down attention processes on early sensory processing [Hillyard and Anllo-Vento, 1998]. Of particular interest, the P2 has also been associated with a word repetition effect [Luck, 2014, Molfese et al., 2005] where the P2 showed a reduced positivity (i.e., a larger negativity) to primed versus unprimed targets. Word repetition is frequently used as an investigative tool in psycholinguistic and memory research. It is a simple empirical procedure which demonstrates that subjects are usually faster in their response to the second presentation of words than the first; such responses may be captured via reaction time (RT) measures across a variety of experimental paradigms such as lexical decision or semantic categorization.

Prior research in which participants read short texts containing repeated words has found three distinct ERP components to be sensitive to repetition: a positive component peaking around 200 ms post-stimulus, a negative component at 400 ms (N400) and a later positivity [van Petten et al., 1991]. However, van Petten et al. [1991] note that the early P2 repetition effect has not been consistently found in other studies, at times appearing with an opposite polarity. Due to the paucity of research using real-time speech signals and the conflicting early results cited, it appears that the processes which control this early component are not well-understood. Among the research questions that remain open are to what extent does deeper lexical processing and explicit memory influence the word repetition effect and what particular cognitive processes elicit this effect? While this investigation did not set out to explore these questions in depth, we address some of them in the context of our results.

8.1.2 Human Experiments and ERP Data

8.1.2.1 Empirical ERP Data

We collected ERP data from 12 Native English speakers from Tufts University (mean age 19.6, 7 male), of which 2 were excluded due to excessive ocular artifacts. All participants self-reported as monolingual and right-handed [Oldfield, 1971], with normal or corrected-to-normal vision/hearing and normal neurological profile. Participants provided written informed consent and were monetarily compensated, as approved by the Tufts University Institutional Review Board.

8.1.2.2 Materials and Design

During ERP recording, participants completed a dual-task paradigm with a primary task of playing a video game (i.e., "Breakout": breaking pre-arranged blocks by bouncing a ball from a controllable paddle) and a secondary task of listening to words through a set of headphones. The dual-task paradigm was important for our ERP modeling task because we attempted to reduce any explicit episodic memory effect so that we could focus on more implicit repetition primary effects by introducing the primary task of playing a video game. For the primary task, we utilized a JavaScript variant of Breakout. Three game levels were chosen based on pilot results, indicating them to be similar in difficulty. For the secondary task, a female experimenter recorded 300 monosyllabic English words to be used in stimuli generation. These 300 words were split into two lists (of 150 words each) matched for psycholinguistic properties (e.g., bigram frequency, length, phonological and orthographic frequency, familiarity, and concreteness). An additional list was created from the two split lists (half from each) so that a total of three lists of 150 words were created. From each of the 3 lists, 50 of the 150 words were randomly selected to be repeated so that each list contained a total of 200 words. None of the repeated words were redundant across lists.

8.1.2.3 EEG Recording

Participants engaged in the dual-task paradigm in a dark, sound-attenuated room while their EEG was recorded using a 29-channel electrode cap. Loose electrodes recorded from 1) below the left eye (LE) to monitor for blinks and vertical eye movements, 2) at the right temple (HE) to monitor for horizontal eye movements, and 3) behind each mastoid (left: A1, right: A2) for referencing (A1) and monitoring differential mastoid activity (A2). Electrode impedances were kept under 5 kΩ for all scalp electrodes, 10 kΩ for both eye electrodes, and 2 kΩ for both mastoid electrodes. We sampled the EEG at 200Hz while an SA Bioamplifier (SA Instruments, San Diego, CA) amplified the signal with bandpass of 0.01 and 40 Hz.

8.1.2.4 Experimental Results

Averaged ERPs were formed for each spoken word (using -100 and 0 ms baseline) after artifact rejection (15.67% of the trials were rejected due to ocular artifacts) and collapsed into conditions (first presentation or repeated) for comparison. The ERPs were then low-pass filtered at 15 Hz. Individual participant ERPs were then averaged into a grandmean of 10 participants, allowing for the analysis of overall auditory language processing effects. Of particular interest is the repetition effect on particular ERP components such as the P2 [van Petten and Kutas, 1991, Rugg, 1987] with an anterior scalp distribution, sensitive to lexical processing and implicated in word recognition processes [Dambacher et al., 2006]. Such repetition effects manifest in the form of attenuated amplitudes to repeated items compared to their first presentation, reflecting the ease of processing for the former relative to the latter. Results indicate the presence of a P2 repetition effect, seen clearly in anterior electrodes between 200 and 400 ms (Figure 8.1).

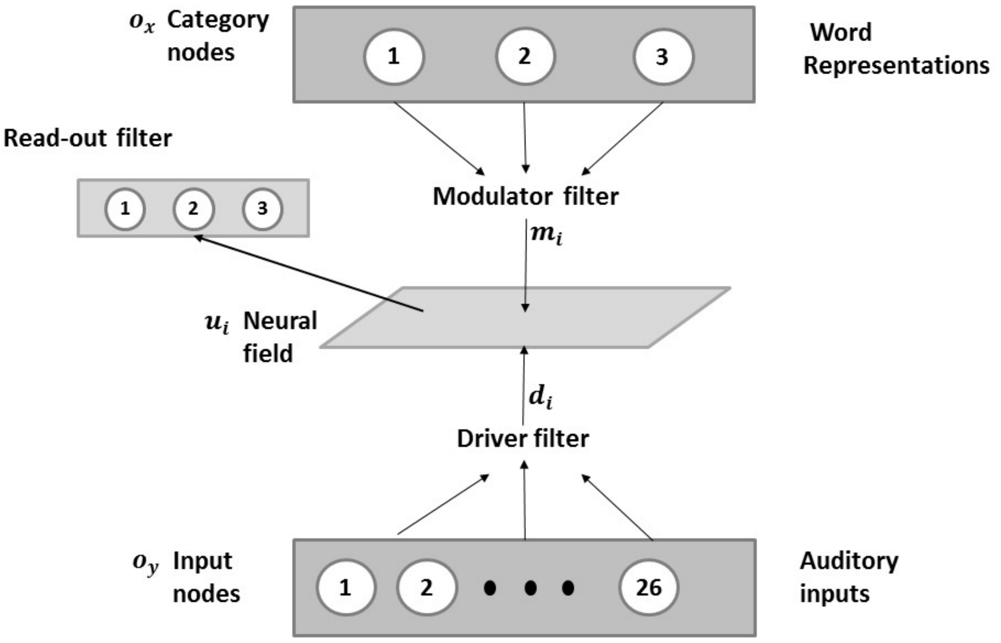


Figure 8.2: Neural field training. The training vector at the word representation layer develops an input signal $s = m_i$ through the modulator filter to each processing unit u_i in the neural field as a random sound exemplar of the same training vector category is played to the input nodes.

8.1.3 Neural Field Model Description

We modeled a single layer of the hierarchical process generally regarded to represent the architecture of speech perception [Grossberg, 2005, McClelland and Elman, 1986, Norris, 1995]. In Figure 8.2, the model architecture consists of (1) a vector of *auditory input nodes*, (2) a vector of *category nodes*, (3) a grid of processing units called a *neural field*, and (4) three fully connected sets of weights to be trained called *adaptive filters*. The field processing units are reciprocally connected to each other through non-adjustable weighted connections using an on-center, off-surround "Mexican hat" distance function [Brady, 2014]. The input nodes carry sensory information which is refreshed with new data at each time step. This input is passed through a "driver" filter to develop a bottom-up input signal to the field. The category nodes carry persistent labeling information which is passed through a "modulator" filter to provide a top-down input signal to the field. The labeling information is also used as the training target for a "read-out" filter.

A neural field in our model is a "sheet" of processing units. If given no input and random initial conditions, all units of the field are guaranteed to quickly fall into a stable equilibrium state with respect to each other such that the entire field may be considered to fall into an equilibrium. Different equilibrium states of the field are associated with different input patterns. The field is updated once every 10 ms (i.e., a time step) using Equation 8.1 which computes the change in its activation. This general equation and its variations are widely used in dynamical systems models, [Amari, 1977, Beer, 2000, Brady, 2014, Grossberg, 2005, Hopfield, 1982, Schöner and Spencer, 2015].

$$\dot{u_i} = -u_i + s_i + h + n + \sum_j \lambda(i,j) \cdot \sigma(u_j) \tag{8.1}$$

The change in activation of a unit, u_i at a given time step is computed as the sum of influence to the unit at that time step minus the activation of the unit from the previous time step. Influence to a unit at a time step comes from an input signal, s_i, the field's slightly negative bias, h, a noise term, n, and from other units within the field. Influence from other units within the field is computed to be the sum of the squashed activations of neighboring units multiplied through corresponding within-field connection weights w. A stepwise squashing function, σ, is used such that only units with non-negative activations can influence their neighborhoods. Within-field connection weights are specified as on-center off-surround by a Mexican hat weighting function, $\lambda(D)$. Input to the function D is the Euclidean distance between two units, u_i and u_j; the output of the function specifies their connection strength.

8.1.3.1 Neural Field Learning

We implemented a learning mechanism in which the driver and modulator filters are trained together that works as follows. The filter weights are initialized with random values which are then updated across training cycles. A training cycle consists of iterations in which the neural field is initialized with random unit activations simulating the passage of time between learning patterns. Then, a training vector is used to generate an input signal s_i through the filters to each unit of the neural field using Equation 8.2, and a random sound exemplar of the same category as the training vector is played to the input nodes as time unfolds. In our experiment, the training vector represents a monosyllabic word. Here, o_y is the activation of a category node, o_x is the activation of an input node, and g_1, g_2, g_3 are gain terms; $\dot{d}_i$ is the change in activation of the driver signal, $\overline{u}_i$ is the running average of the unit being updated, and $\overline{m}_i$ is the running average of the modulator signal to a unit.

$$s_i = g_1|\dot{d}_i| \cdot (g_2\overline{m}_i - g_3\overline{u}_i) \tag{8.2}$$

$$m_i = \sum_y w_{iy} \cdot o_y$$

$$d_i = \sum_x w_{ix} \cdot o_x$$

The weights of the modulator and driver filters are adjusted following Equation 8.3, a variant of the delta training rule.

$$\Delta w_{ix} = \eta \cdot \bar{o}_x \cdot (\overline{u}_i - \dot{d}_i) \cdot |\dot{u}_i| \tag{8.3}$$

$$\Delta w_{iy} = \eta \cdot \bar{o}_y \cdot (\overline{u}_i - \overline{m}_i) \cdot |\dot{u}_i|$$

Learning proceeds as the training vector persists for the duration of the input sound as the neural field adjusts itself in response to its input, updating the modulator and driver filters at each time step. Subsequently, a new iteration begins by initializing the field to a new random state and associating the transformation of that state through time with the next input training vector (i.e., new word), and so on. A cycle is completed when all training vectors have been exposed to the model in random order, at which point a new training cycle begins.

In Equation 8.3, η is the learning rate, $(\overline{u}_i - \dot{d}_i)$ and $(\overline{u}_i - \overline{m}_i)$ are the errors to be minimized; cyclic training continues until the learning error is reduced to asymptote. The last term of the equation, $|\dot{u}_i|$, is an innovation which allows learning to occur only if there is a change in the target neural field and therefore important associations are maintained even as learning proceeds over time.

8.1.4 Experiment 1: Modeling Word Repetition

The model's read-out filter is trained in order to evaluate how well the neural field categorizes its input. The weights of this read-out filter are updated using the "delta rule" as in Equation 8.4. Training vectors o_y are converted to target vectors T_y by setting the negative values of the training vectors all to zero. The generated output is notated as $\hat{o}$.

$$\Delta w_{yi} = \eta \cdot u_i \cdot (T_y - \sigma(\hat{o}_y)) \tag{8.4}$$

Where:

$$\hat{o}_y = \sum_i w_{yi} \cdot m_i$$

We selected a subset of five monosyllabic words from the stimuli used in the empirical experiment: "beach", "dog", "soup", "bog", and "tend". Four exemplars of each word were recorded separately by a male speaker as male voices span a lower frequency range making for easier speech processing by the model. The recordings were transformed into the 26 coefficients shown in Figure 8.2 and were provided as input to the model in 10 ms time steps. The model was trained on three target words from this set, “beach”, “dog”, and “soup”. How well the model learned was measured by computing the error as the sum of the differences between “readout” vector generated as output by the model and the corresponding target word.

8.1.4.1 Modeling the ERP Measure

We chose to model the ERP as the difference between the modulator signal and the field activation. This can be thought of as analogous to error values or implicit prediction error. Implicit prediction error at multiple levels of language processing is thought to play a critical role in language comprehension [Kuperberg and Jaeger, 2015]. Within probabilistic frameworks, implicit prediction error has been linked to other language-related components such as the N400 ERP Kuperberg [2013], Xiang and Kuperberg [2014], Kuperberg [2016], as well as non-linguistic ERP components [Friston, 2005, Wagongne et al., 2005]. Moreover, the N400 ERP component has recently been simulated as cross-entropy error at a semantic level within a connectionist model [Rabovsky and McRae, 2014a].

The ERP at time t is computed as shown in Equation 8.5; m_i and u_i are each unit’s modulator and field activation respectively:

$$ERP_t = \sum_i |m_i - u_i| \tag{8.5}$$

8.1.4.2 Modeling Results

The words from the test input were presented in the following order: “soup”, “dog”, “dog”, “dog”, “beach”, “dog”, “bog”, “tend”. The neural field was trained on “soup”, “dog”, and “beach”; “bog” and “tend” were novel stimuli the field was not trained on. Figure 8.3 shows that the model replicates the repetition effects, i.e., the maximum ERP values at a t after the first exposure of the word “dog” are all smaller than the first peak, until a different word is presented. At this time, the neural field is perturbed into a different state, releasing it from the effect. A subsequent presentation of “dog” no longer elicits a repetition effect, producing a

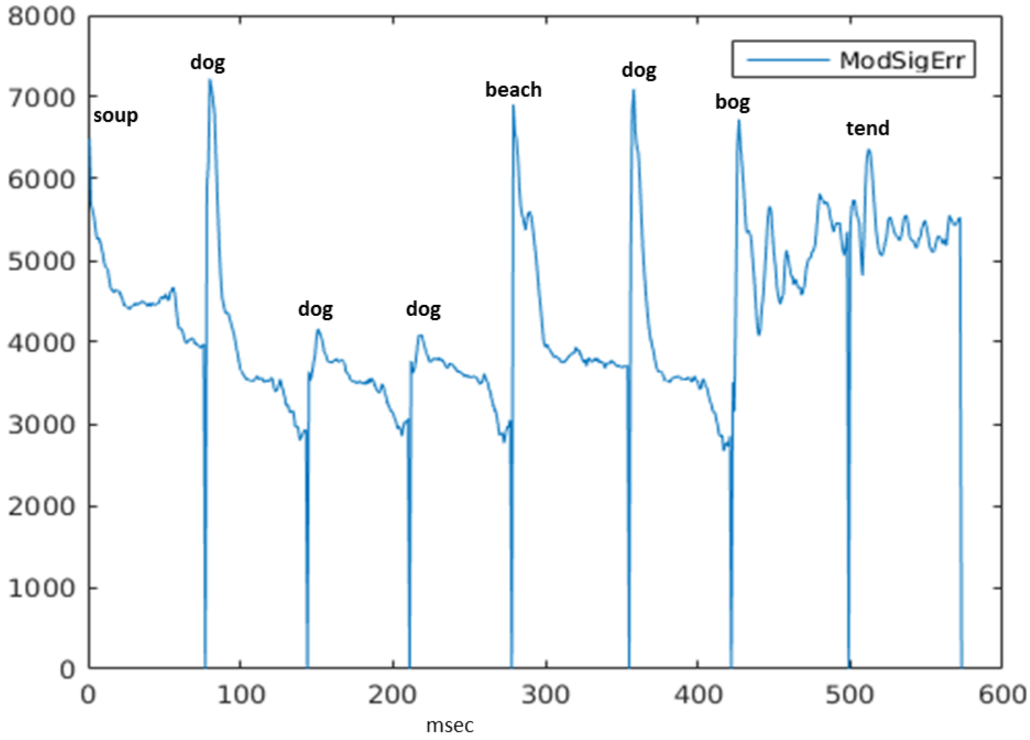


Figure 8.3: Modulator-Field difference for repetitions of the word "dog"

larger peak as the field resettles into the equilibrium state for "dog". In Equation 8.5, the modulator signal, m_i, can be thought to "predict" the next equilibrium state the neural field u_i is likely to settle to. This suggests that a smaller amount of perturbation is required to "nudge" the settled field into a new equilibrium state upon presentation of a repeated word. The presentation of untrained, novel stimulus, i.e., "bog" and "tend", does not show the repetition effect as these words are not predicted by the modulator signal.

Table 8.1: Model Fitting

Interval Width	Model Proportion	ERP Data Proportion
100 ms	1.41	1.60
112 ms	1.53	1.53
120 ms	1.66	1.58
Best fit (112 ms) 144 ms - 256 ms		

We note that model does not aim to fit the polarity of the P2 ERP as what gives rise to the polarity is not well-understood and as there have been inconsistent reports on the word repetition effect as mentioned earlier [van Petten et al., 1991]. Furthermore, it is the nature of ERP measurement that the interval within which a given effect is manifested varies somewhat between experimental paradigms. However, the model should fit the magnitude and the duration of the human ERP data. Thus, to compute the model fit, we looked at the ERP data intervals centered around 200 ms as this interval contains the P2 effect and computed the proportion as follows. We took the area under the ERP curve within an interval for the first presentation of the word "dog" and divided it by the identical interval contained under the repeated presentation to calculate its proportion. Referring to Figure 8.3, we also took the area under the ERP curve generated by the model and performed the same calculation. As shown in Table 8.1 we found that the 112 ms interval around 200 ms (i.e., from 145 ms to 255 ms) showed both proportions to be identical i.e., 1.53, thus demonstrating it is possible to find a good model fit to the experimental data.

8.1.4.3 Discussion

We designed our model to be a single neural field reflecting processing in the auditory cortex and hypothesized that this forms a "layer" of phonological processing. In order to provide a modulator signal, we simulated the existence of a deeper word-form layer by "clamping" the modulator signal to the three words the model was trained on (i.e., "beach", "dog", "soup") and this was fed "down" to the neural field as its modulator signal. We did not presuppose which ERP correlates would occur using only one neural field layer and did not set as a goal to identify all possible auditory effects; we were not concerned with capturing non-speech auditory processing at all.

The model succeeded in capturing the repetition effect noted in the experimental results as can be seen in Figure 8.1, most notably in the central scalp ERPs e.g., Cz. Figure 8.3 shows a diminished response to the initial presentation of the word "dog" at 75 ms with the repetition effect occurring at 150 ms and 225 ms. Note that the typical convention is to plot the ERP, with the area above the x-axis as negative and the area below as positive. Thus the model and ERP waveforms covary in amplitude and polarity with the repetition (i.e., in the model the repetition effect is "more negative" than the initial presentation).

The model demonstrated the immediate word repetition effect using a single neural field sheet, without modulator input from deeper lexical and semantic processing layers. This suggests that the ability of a single neural field layer to learn sound patterns (i.e., phonemes, monosyllabic words) alone appears to be sufficient to account for the immediate word repetition effect and the release from repetition. We believe this to be among the first computational models to match the time course of ERP events on real-world, real-time data, and the first model to do so using spoken word perception i.e., we used the same data that was presented to the experiment's participants and validated the model fit. These results suggest that our neural field approach can now be used to build additional layers and thus model later ERPs.

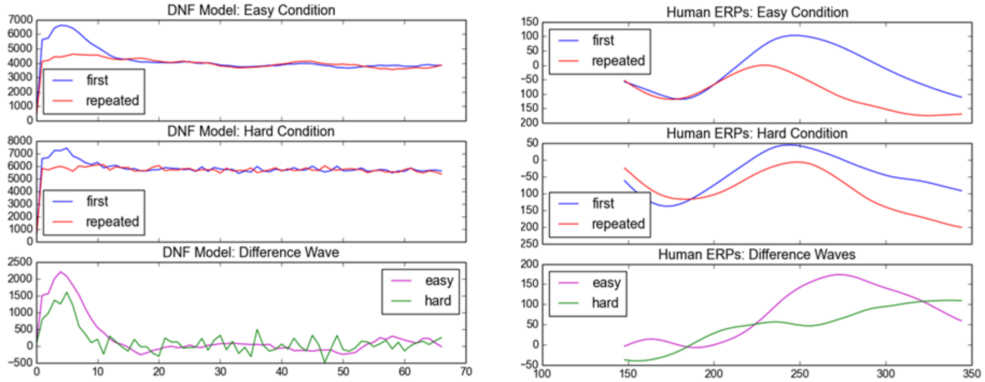


Figure 8.4: Model fitting. For the Human ERP and for the neural field model, the difference between the first and repeated presentation of the words was computed for each condition and the area under the curve (AUC) was calculated. The AUCs of Human ERP and neural field model difference waves were compared to determine how well the model fit the human data.

8.1.5 Experiment 2: Modeling Task Difficulties

As described in Section 8.1.2.2, the human experiment was a dual task paradigm with concurrent EEG recording. Here we focus on the primary task in which participants played a block-breaker game with three varying difficulties (easy, medium, hard). The model was trained as Experiment 1 (see Section 8.1.4.1), to learn the words "beach", "dog", and "soup". We ran the model under two work load condition: easy and hard, using the same sequence as in Experiment 1: "soup", "dog", "dog", "dog", "beach", "dog", "bog", "tend".

As shown in Figure 8.4, we computed the difference between the first and repeated presentation for each condition and used this "difference wave" to compare the model and the human data. We computed the proportion of the two AUCs separately for the neural field model and the human data. If the model responds similarly to the human ERP, then the corresponding AUC proportions ($easyAUC/hardAUC$) should be similar. We found a very good fit to the human data, where the proportion for the neural field model = 1.63 and that of the Human ERP = 1.65.

8.2 A Neural Field Model of Sequence Perception

We show how temporal and spatial information can be represented as stable patterns in a dynamical system. We describe a model in which category perception arises from the incremental recognition of temporal patterns from sequences of inputs and this is accomplished by decoding a pool of recurrently connected artificial neurons which is called a neural field. In an example application, we use these patterns to identify a set of words which share the word onset represented by the input sequence, consistent with the Marslen-Wilson COHORT model of word recognition. Similarly, we evaluate the extent to which information contained in the bottom-up sensory signal can be used to determine word boundaries. We suggest it is plausible that a neural field offers a naturalistic explanation of how perception arises in word processing.

8.2.1 Introduction

The brain encodes and processes sensory input acquired from the environment. Sensory input, regardless of modality, is encoded as spatiotemporal patterns, and a superior form of pattern processing has evolved in humans coinciding with the expansion of the neocortex. In this brain structure, several essential cognitive processes such as visual, auditory, and speech perception occur [Koch, 2004, Mattson, 2014]. These processes include not only recognizing patterns, but also classifying them [Grossberg, 2005]. During this processing, different sensory inputs which represent members of the same category are mapped to a singular representation for that category. In speech processing, for example, all pronunciations of the phoneme “ə”, are mapped to the same pattern, allowing for invariance in speech perception across multiple speakers [Kleinschmidt and Jaeger, 2015]. Consistent with these hypotheses, our model uses patterns of activation to represent sequences of states in the context of perceiving words; we modeled these states as equilibriums in a *neural field*.

The human neocortex consists of six layers of tissue containing approximately 10^{10} neurons. Columns of tissue can be represented mathematically as neural fields, which form patterns of activation through interaction with each other [Amari, 1977]. These interactions between fields generate patterns of activation in a fashion that is believed to be similar to how sensory information is represented in the human neocortex [Amari, 1977, Brady, 2012]. These patterns represent an encoding of spatial and temporal information from the brain’s sensory input stream.

Each neuron in a neural field F (Figure 8.5) is connected to each of its neighbors with weights that create an on-center off-surround activation pattern, where the closest neighbors provide a positive influence on activation, further neighbors a negative influence, and the furthest no influence. If given no input and random initial conditions, the units of the field are guaranteed to quickly fall into a stable equilibrium state. Different equilibrium states of a field can be associated with different inputs, and thus the states of activation in a neural field can be used to store information by associating them with category labels as we demonstrated in Section 8.1.

In this investigation, we demonstrate a model of word perception using neural fields. Our research is not focused the initial interaction between perceptual signals and the sensory apparatus. We are instead interested in the processing of the output of such apparatuses, and how it can be used to constrain the patterns of activation in higher level cognitive processes, like lexical representation. Our model uses two neural fields, each representing a level of cognitive processing. Since sensory information unfolds over time as a continuous sequence, the input presented to the first neural field is a sequence of feature vectors which represent the letters of an artificial font. Sequences of output features from the first field representing letters are then presented as input to the second field which identifies likely word boundaries and classifies these letter sequences as words.

There are many theories about how patterns of activation in the lexicon are formed once the sensory information has been received [Dahan and Magnuson, 2006]. This work focuses on the Marslen-Wilson [1987] COHORT model, which theorizes that information contained in the bottom-up perceptual signal can be exploited to determine which lexical items should be activated, and also used to identify perceptual characteristics such as word boundaries. To explore the extent to which this information is sufficient, we have developed a model where word onsets constrain the set of activated lexical entities such that word onsets activate lexical items with shared onsets. Our model thus makes predictions similarly to the COHORT model; the initial information contained in the sensory signal influences the activation of an initial word-cohort, allowing it to predict word boundaries in a higher level of processing.

8.2.2 Representing State with a Neural Field

Our model is composed of two layers of neural fields. The structure of a single layer is shown in Figure 8.5. An Input vector (I) is fully connected to the neural field (F) by input weights (W_i). F is fully connected to an output vector (O) by

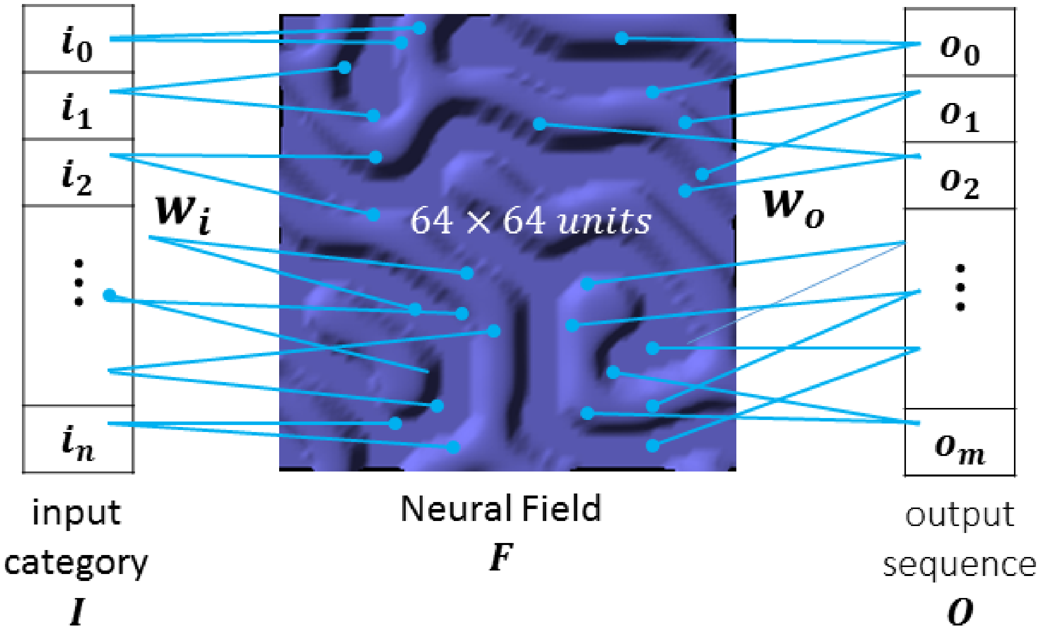


Figure 8.5: Single field design. Note: I and O are fully connected to F, but only a few connections are depicted here, for clarity.

output weights (W_o). In our model, such a layer (input, field, and output) can be interpreted as a cognitive processing layer, computing a specific function such as letter or word detection. These layers can be combined to represent a hierarchy of cognitive processes shown in Figure 8.7.

Our model is based on the following principles of dynamic field theory. Patterns can be stored as stable equilibrium states. A sequence can be "remembered" as a unique equilibrium, unrelated to any previously generated equilibrium, by calculating the sum of the pattern generated by the current input and the pattern representing the previous sequence. Fields converge to a stable equilibrium state after applying a finite series of "settling" operations after which the field ceases to change. Finally, fields can be forced out of a stable state into a target state by applying a finite series of operations which includes the target as its field input.

8.2.2.1 Field Dynamics

The input to a layer is received as a vector I, whose dimensionality is the number of discrete categories in the input domain. This input is used to first calculate the

$n \times n$ matrix F_t (in our evaluation $n = 64$), which represents field activation at the given point in the sequence of input vectors. F_t is calculated using the following equations which are a variation of those widely used in dynamical systems [Amari, 1977].

$$D = W_i I \tag{8.6}$$

$$\sigma(x) = \begin{cases} \dfrac{x}{x+1} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{8.7}$$

$$S = W_{mh}\ \sigma(F_{t-1} + D) \tag{8.8}$$

$$F_t = \sigma(S + h + n) \tag{8.9}$$

As shown in Equation (8.6), the driver input, D, is generated by multiplying the input vector I by the input weights, W_i (dimensionality $n \times n \times \parallel I \parallel$). D is then added to the current field equilibrium, F_{t-1}, and the result is squashed to the range $[0, 1]$ using Equation (8.7). This new equilibrium represents the sequence of input seen up to time t, plus the input at t. The result is multiplied by the within field weights, W_{mh}, which are defined using the Mexican hat function;[1] the result is the field influence term, S, Equation (8.8). Small bias h and noise n terms are added to the field influence, and the result is squashed again to produce the field activation F_t, Equation (8.9).

8.2.2.2 Settling to an Equilibrium State

Once a field has been updated from an input, a settling operation is applied resulting in convergence to an equilibrium state. This process is expressed in Equation (8.10), which is based on Equations (8.8) and (8.9) with the notable exception that the value of the field at the previous time step is not added to the input. This operation is repeated until a stable equilibrium is reached, determined by comparing the field

[1] A Mexican Hat function where D is the Euclidean distance between two units in the field: $W_{mh} = e^{\frac{-D^2}{r^2 d}} \cdot (cos(\frac{\pi D}{2r}) - z) \cdot (\frac{1}{1-z}), r = 4.5, z = 0.15$

activation at time t with its activation at $t-1$, repeating until the difference is below a small epsilon value.

$$S = W_{mh}\ \sigma(S + n) \tag{8.10}$$

8.2.2.3 Layer Output

The settled equilibrium is used to calculate the layer's output vector, O, whose dimensionality is the number of categories in the output domain.

$$O = tanh(F_t W_o) \tag{8.11}$$

The state of the field is multiplied by the output weights W_o $(n \times n \times \parallel O \parallel)$, and their product is passed through the hyperbolic tangent activation function. The result is a vector whose values are normalized to the range $[-1, 1]$.

8.2.3 The COHORT Model Using Neural Fields

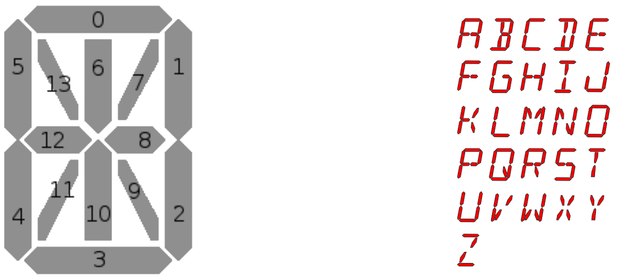
The Marslen-Wilson COHORT model of spoken word recognition suggests that the real-time constraints of a speech signal influence how bottom-up information is used to determine which items in the mental lexicon become activated. According to the model, on each new input onset only the cohort of possible values remains activated; a cohort is the set of all lexical items that share an onset. A decision is reached only when one possible value remains in the cohort. Consistent with the original version of COHORT (which assumed a highly categorized, abstract string of phonemes rather than feature vectors as input), our model uses only the features present in the bottom-up sensory input to develop the cohort.

The neural fields in our model simulate what happens when sensory information makes contact with the mental lexicon. It is assumed that, once past the sensory apparatus, information flow through differentiated neural architectures, e.g., visual or auditory, is represented in the same fashion. In our evaluation we chose to focus on visual information, and the following sections describe how it is handled by the model.

8.2.3.1 Model Input

The input to our model is a sequence of feature vectors which represent the output of the visual perceptual system. As in the Interactive Activation model of word recognition [McClelland and Rumelhart, 1981b], visual features are extracted from the raw input, sequences of letters, by separating each letter into a set of component features. These features can be thought of as the pen strokes used to write the letter. For simplicity, we have chosen the font used by Rumelhart and Siple [1974] which is shown in Figure 8.6. Sequences of feature vectors are generated from a letter by arbitrarily circumnavigating the font clockwise from the outermost feature, spiraling inward. For example, the letter "R" is represented by the sequence [0, 1, 4, 5, 8, 9, 12].

In our implementation, the model receives visual features as input. As mentioned earlier, the input features could also represent information from other sensory modalities. For example, the input features could also come from a speech recognizer and instead represent sequences of phonemes. At the cognitive processing level, the model's basic results do not depend on what the input represents. With different input features there are obviously low-level feature processing issues (e.g., different types of variance in the input) which are outside the scope of this work.



(a) Labeled Segments

(b) Letters

Figure 8.6: 14-segment display using the letter font.

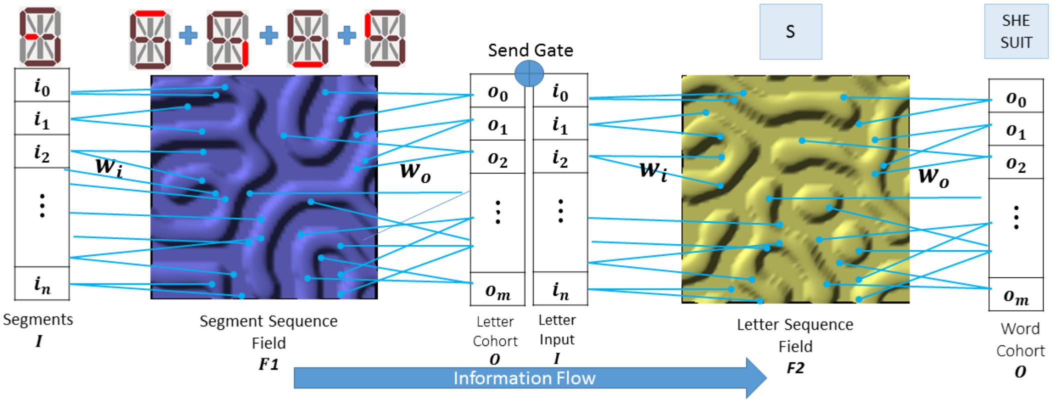


Figure 8.7: Neural Field based Cohort Model architecture.

8.2.3.2 Model Architecture

The architecture of our model is shown in Figure 8.7. It uses two neural fields, each representing a stage of cognitive processing: *F1* which is a letter detector, and *F2* which is a word detector. $F1$ and $F2$ are connected via a Send Gate which controls when information from $F1$ is sent to $F2$.

Before the model can be used, it must first be initialized. This initialization generates the associations between input and neural field equilibrium states which are used to detect sequence boundaries. During this initialization the model is presented with sequences of input features and the boundaries between them that represent meaningful units. Perceptrons are trained to detect these boundaries based on the equilibrium states which represent them.

Once initialized, information flows through the model as shown in Figure 8.7. Letter segment sequences previously generated by the visual recognizer flow as input to $F1$; letters detected by $F1$ flow as letter sequences to the word detector $F2$. Perception arises as the generated pattern predicts a set, or cohort, of letter or word candidates. As new features are fed incrementally into the model, a new pattern is generated and each field's perceptron updates its prediction, removing candidates from its cohort. Letter or word recognition occurs when there is a single candidate left in the respective cohort. When recognition occurs, the send gate is opened sending the output perceptron's value to the next layer as input. In the case of the letter layer, the output is sent as input to the word layer; in the case of the word layer, the output value of the perceptron is used by the decoder perceptron to generate results interpretable by a human.

8.2.3.3 Model Initialization

The characteristic theory of the COHORT model is instantiated in the neural field model during initialization. This initialization forms the associations between input features and neural field equilibrium states used during perception. This initialization is composed of three steps: (1) Initial equilibrium generation (2) W_i training (3) W_o training.

The initial values of the weight matrices used in the model ($F1$: W_i, W_o and $F2$: W_i, W_o) are chosen randomly from a truncated normal distribution with a standard deviation of: $\frac{2}{\sqrt{n_{inputs}}}$ Using this particular standard deviation helps the training to converge more quickly [Géron, 2017].

Initial Equilibrium Generation: A "seed" equilibrium is generated to represent each unique input feature a field will receive. For the letter detector, $F1$, 15 equilibriums are generated to represent each of the 14 possible letter visual features, plus an equilibrium to represent the beginning of a sequence when no input has been presented yet. For the word detector, $F2$, 27 equilibriums are generated, for the 26 letters in the English alphabet and one more for the initial sate. These initial equilibriums are generated by a variation of Equation 8.6 where I is the product of a one-hot vector whose 1-bit corresponds to the ordinal value of the letter segment in the range $[0, 15]$ or letter in the range $[0, 27]$ and the randomly drawn W_i for the given field.

W_i **Training:** For a feature vector (i.e., letter segments for the first field, letters for the second), a set of weights (W_i) is trained which will reliably reproduce the initial equilibrium associated with that feature vector. The model's operation assumes that a *settled* field generated from new input can be added to the current settled field to produce a new equilibrium representing the input sequence seen thus far. Without trained driver weights, an *unsettled* equilibrium would be added, violating a core model assumption. The training of W_i, uses a version of the perceptron learning rule, Equation 8.12, to train the single layer perceptron whose activation is found by multiplying the driver input by W_i,

$$\Delta W_i = \eta I(Target - IW_i) \tag{8.12}$$

where $Target$ is the seed equilibrium for the category and η is a learning rate. Training proceeds until $Target - IW_i < 0.0001$. This approach is a variation of Hebbian learning, a biologically plausible mechanism for learning associations between neurons [Laszlo and Plaut, 2012a].

W_o **Training:** The output weights W_o map a field's current equilibrium to the output domain relevant to its cognitive layer (i.e., letters cohort or words cohort). For a given cognitive layer the output vector O represents the members of the cohort that are currently active. O is the size of the lexicon of known labels at the given cognitive level, each of its elements representing a member of the lexicon. A member of the lexicon is considered activated if the value of its corresponding element in O is above an activation threshold. For example, the letter segment sequence [0, 1] is the prefix of the letters {A, B, D, O, P, Q, R} (see Figure 8.6a).

The weights W_o are trained so that the same value of O can be calculated every time a corresponding equilibrium is present in F. The weights are updated using Equation 8.13,

$$\Delta W_o = \eta F(Target - FW_o) \tag{8.13}$$

where $Target$ is the vector in the output domain (e.g., letters or words) indicating cohort membership of the lexical entries whose onset is represented by the equilibrium of the neural field F. Training proceeds until $Target - FW_o < 0.001$.

8.2.3.4 Detecting a New Sequence

An equilibrium for a sequence is generated by adding the seed equilibrium of the new element of the sequence to the current sequence equilibrium. For the first segment in a letter, its seed equilibrium is added to a default value (i.e., the 15^{th} seed equilibrium). This process is repeated for each segment of the letter and after each addition, the field is settled. A challenge in implementing the COHORT model is detecting when a new sequence begins. In our model, an input sequence is "remembered" as an equilibrium whose value is the sum of the equilibriums seen as input thus far. Thus, when the end of the sequence is detected there must be some way of resetting the field so that the next sequence is not affected by the previous sequence. To do this, each field has a reset gate whose purpose is to detect the conditions under which the field should be reset to its default equilibrium.

The default equilibrium is the starting state to which subsequent equilibriums are added. The model hypothesizes that a reset signal represents constraints arriving top-down from higher cognitive processing levels (e.g., syntactic, semantic, pragmatic) as well as bottom-up from the features contained in the input data.

8.2.3.5 The Send Gate

Each layer of the model has an associated Send Gate which controls the information that it sends to the next highest level of cognitive processing. The first layer's send gate connects the letter detector field to the word detector field and the second layer's determines the overall output of the model. In different configuration of the model, the second layer's send gate could connect to a third field and so on. Send Gate processing is the same for every layer (refer to Figure 8.8). First, the input features A are presented, and the field is updated B. The cohort is then calculated C and evaluated by the Reset Gate D. The field may or may not then be reset to its default state. The cohort is calculated and if it has shrunken to one member, the Send Gate E opens.

Notice that the Send Gate only opens when the cohort has shrunk to one

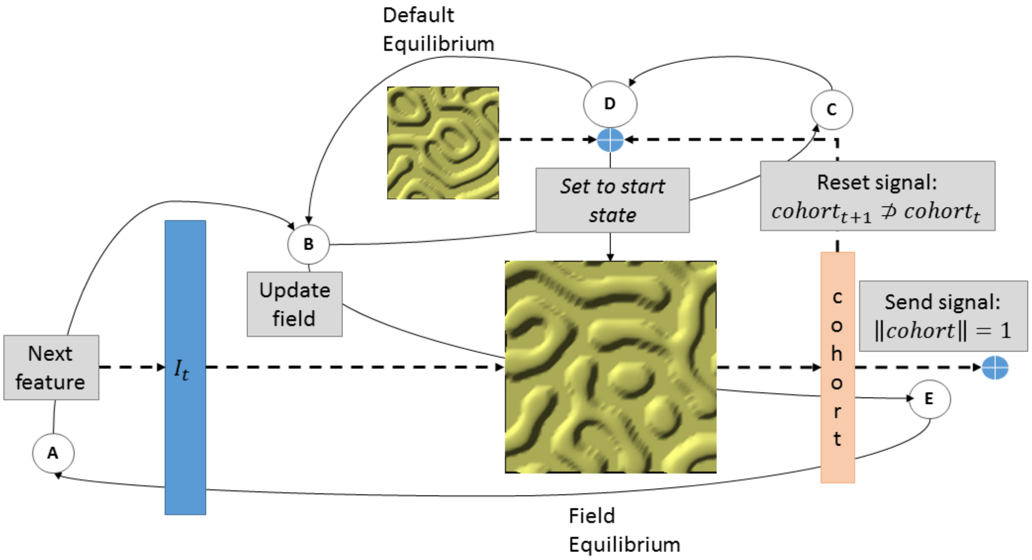


Figure 8.8: Reset/Send Signal Processing. If the reset gate (D) is open, it will set the field to a default equilibrium; otherwise it will update the field to the sum of the equilibrium of the current field and the equilibrium of the new feature. Send gate processing (E) takes place *after* the reset gate is processed.

member. Thus we must ensure that the state of the cohort is reset so that a new sequence can be subsequently recognized,otherwise a feature that is repeated across category boundaries will not be recognized. The Send Gate behavior models the recognition point prediction of the COHORT model which states that word recognition occurs as soon as sufficient information is received such that all other candidates are eliminated [Marslen-Wilson, 1987].

8.2.4 Model Evaluation

The primary goal of our research was to determine whether neural fields are a plausible way to model word perception. Prior research theorizes that humans represent word forms as categories, abstracted away from variability [Dahan and Magnuson, 2006] and it is this view that our model seeks to explore. There are several well-known cognitive models (e.g., COHORT, TRACE, Neighborhood Word Activation) whose theories make different predictions once the input signal make contact with the lexicon. We chose the COHORT model as a starting point and explore whether two of its predictions can emerge from our neural field model: word-initial cohort and the identification of word boundaries. The operation of the model is summarized as follows:

1. Each feature (i.e., letter segment) of sensory input is converted to a pattern.
2. Sequences are generated by adding the current input's pattern to the previous input's pattern.
3. Perception arises as the generated pattern predicts a set, or "cohort", of letter or word candidates. A perceptron is trained to decode the pattern and interpret the prediction.
4. As new features are fed into the model, a new pattern is generated and the perceptron updates its prediction, removing candidates from the cohort.
5. Letter or word recognition occurs when there is one candidate left in the cohort.
6. New categories are recognized at the point when either the cohort is empty or when new candidates are added.

8.2.4.1 Materials

The TIMIT corpus [Garofolo et al., 1993] provides a set of 10 phonetically rich sentences spoken by 630 speakers of eight major dialects of American English which are annotated at the word and phoneme level. The annotations of the corpus were used as a set of naturally occurring sequences to train the model's letter and word detectors. The text of the corpus was used to create feature vectors, as described in *Model Input*, which was presented to the $F1$ as a sequence of letter segments.

8.2.4.2 Results

The entirety of the TIMIT training set was pre-processed by the visual feature recognizer and its output, an unbroken sequence of letter segments was presented as input to the model. In the first experiment, the model was artificially reset to a default state at the end of every word so that errors in the perception of one word did not affect the perception of other words. This was done to verify correct operation of the model. For 100% of the words in the lexicon, the activation matched the ground truth for every letter segment in that word. Furthermore the model generated the correct cohort (when one existed) of letters for every letter segment sequence and of words for every letter sequence. In a separate experiment, the model was not reset and in 82.5% of cases, the model detected the word level transition, suggesting that bottom-up information alone is insufficient to detect word boundaries.

8.2.4.3 Discussion

The model uses the structure of the data to represent top-down cues which are simulated through a "forced reset" when the start of a new letter or word is detected from the structure of the input data set. This is not ideal but allows the model to continue processing when the bottom-up cues alone are insufficient. One alternative to the forced reset would be to train a detector to recognize likely word boundaries in a training corpus, using it to augment the existing cohort-based reset mechanism. Consider the following sequence of letters without any explicit separation (we could have equally used a letter segment sequence, but that would have been harder to visualize):

> *shewashedyourdarksuitingreasywashwaterallyear*

Humans can usually distinguish each letter sequence of a word and consequently recognize each word of the target sentence; however it is not as straightforward for a computer model to do so. Without further information constraints, a naïve model might correctly reject all sequences of letters that form non-words (e.g., *shew*) but erroneously recognize legal words such as *suiting*, resulting in a syntactically implausible reading of the sentence. Our model attempts to discern sequence boundaries by exploiting the cohort dynamics when processing letters. As a sequence of letters is read into the model, a cohort of possible words is initially formed which shrinks in size until only a single word candidate is left; this is the word's recognition point. If a shrinking cohort begins to grow again when a new letter is added to the sequence,decided this might indicate the start of a new word sequence and that the model should reset the field (D in Figure 8.8).

Since the present design does not model higher level cognitive processes, we abstract over all those that might be relevant to detecting a category boundary and combine them into one signal per field called *forcedReset*. Specifically, the data is preprocessed by the visual feature recognizer so that the letter segments have been grouped into sequences by letter. This roughly corresponds to how the higher areas of the visual cortex constrain lower area feature sequences during perception [Friston, 2005]. The model uses this information to force the letter detector field to reset at the start of every new sequence. Likewise, the model uses the word size as a top-down cue to force a reset in the $F2$ word detector. During evaluation, the percentage of times the model accurately detects a word boundary using only the bottom-up signal is calculated; the forced reset ensures the model can continue processing when there is insufficient bottom-up information.

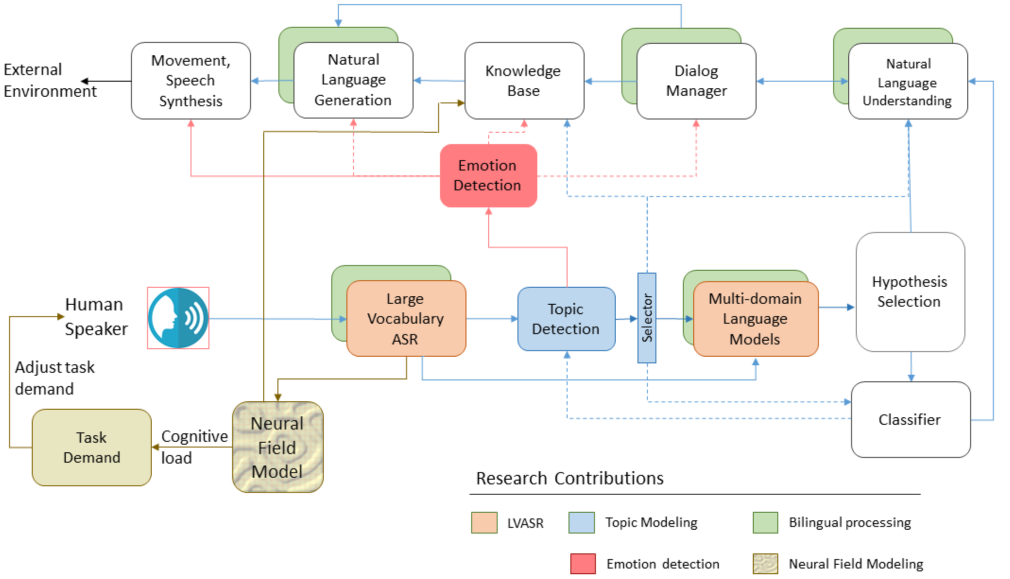


Figure 8.9: Task Demand processing. The neural field received a correlate of human language processing (e.g., an ERP component) and sends it to a Task Demand component. The component analyzes the demand level and adjusts the human task accordingly.

8.2.5 Future Directions

The first version of COHORT assumed input to be an abstract phoneme string. Thus, we arbitrarily chose to present visual input to the neural field as an unambiguous, noiseless sequence of letter segments which made it easier to visualize the model's operation in its graphical user interface. Real-world data is noisy yet perception still arises from these cognitive "noisy channels". Developing a design that incorporates noisy channels is key to understanding situated cognitive processes. Similarly, the input is invariant. In the speech perception domain, humans can usually recognize what is being said regardless of the speaker's accent, gender, etc. The model design needs to incorporate the ability to map varying input to invariant representations in order to simulate human performance in most perception domains. The model uses bottom-up information contained in the input signal to determine word boundaries, which is insufficient for 100% accuracy. Training an additional perceptron on a large speech corpus such as TIMIT should allow the model to statistically learn when a word boundary is likely to occur and this can be as a top-down cue to be added to the reset signal and improve its accuracy.

Lastly, human cognitive language processing in the auditory and visual domains is often studied using electro-physiological measures such as Event-related Potentials (ERPs). As described in Section 8.1, we demonstrated a mapping of a single neural field model's dynamics to an ERP component. In Figure 8.9, we demonstrate how to include a neural field to monitor levels of human cognitive performance and adjust the cognitive workloads as required. To do this, the neural field is connected to the LVASR component from which it receives acoustic features. Assuming, the field dynamics vary proportionally to workload (as shown in Experiment 2 Section 8.1.5), the field will send a differential signal to the Task Demand component. This component will use this signal to analyze the demand level and adjusts the human task accordingly.

8.3 Summary

We developed a dynamic neural field model of phonological processing of monosyllabic spoken words and compared it with a separately designed experiment which measured ERP responses of participants to spoken words. We believe this to be the first model to match the time course of ERP events on real-world, real-time data. We found a good fit between the model and the human ERP data. The model succeeded at replicating the word repetition effect showing a positive correlation with the experiment's P2 measurements. This suggests that a minimal neural field model can perform some components of auditory processing (e.g., detect immediate word repetition) and generate a correlated ERP effect. Future investigations might explore modeling deeper lexical and semantic processing and related mid-to-late ERP effects by connecting additional neural field layers in a hierarchy which will allow feedback from the deeper processes to affect computations at earlier layers.

We explored the cognitive process of word recognition by creating a neural field model of the COHORT theory of Marslen-Wilson [1987]. This theory describes how sensory input is mapped to a specific word from a person's mental lexicon. Whereas Marslen-Wilson predicted the identification of a word cohort from which a unique word is selected and recognized he did not address how it might arise functionally from the input signal nor did he specify an implementation of the model. Moreover, we know of only one implementation of COHORT [Johnson and Pugh, 1994]; it too conceives of encoding the input as patterns from which a cohort emerges and resolves. However it does not discuss the underlying algorithm for this process nor how it was trained, so it is difficult to assess its plausibility. In contrast, the presented model provides a general way to encode sequences in patterns and to find positions within those sequences which is applicable to any type of sensory information unfolding over time.

We have demonstrated that neural field models can be applied to measure at least one level of human performance and can also be connected in layers in a computational model of one psychological theory of language processing. On the basis of these investigations, we have shown how it is possible to connect the neural field to the ASR component of our companion robot framework to monitor the human's performance and provide feedback to other components.

Chapter 9

Conclusion

Humans are great communicators. We use language, facial expressions, tone of voice, and gesture to convey our intentions. From this we can infer another person's intentions and even feel what they are feeling. However, facial expression, vocal tone, or speech itself may not always be a reliable or available mode of communication. They may be true in noisy environments, during fast moving, urgent situations, or in persons whose presentation of physiological condition, such as Parkinson's disease, can damage these communication modalities. If the person's true emotional state is misinterpreted by a caregiver, this often leads to depression. To address this, we developed the situated emotion expression framework which a robot can use to detect emotions in one modality, specifically in speech, and then express them in another modality, through gestures or facial expressions. Introducing a companion robot equipped with this framework into situation in which the person living with PD interacts with a caregiver either at home or in a clinical setting is part of a longer-term goal. In this dissertation, we described the investigations that lead to the development of the emotion detection and expression components and evaluated their performance situated in a robotic cognitive architecture. We demonstrated improved accuracy in the semantic interpretation of user's speech by extending the framework to a multi-level process, in which context analysis can be requested as a means to reinterpret the utterance. Finally we showed that it is possible to extend the framework to a multi-lingual environment and to monitor and adapt to human level performance, based on the results of related investigations.

9.1 Contributions

In this dissertation we have made contributions regarding emotion detection, emotion expression, and the framework as a whole. We will begin by describing the framework contributions and then the related technical work.

9.1.1 Situated Emotion Expression Framework

The situated emotion expression framework consists of large vocabulary automated speech recognizer, the emotion detector, and the emotion expressor. The contribution of this framework includes the following:

- We implemented a Large Vocabulary Automatic Speech Recognizer (LVASR) so that the human can speak to the robot using natural language.

- We designed and implemented a Web-based tool (EMIT) which human evaluators used to obtain the ground-truth emotion valence and arousal values for the training corpus.

- We demonstrated that a generative topic model (Latent Dirichlet Allocation) can be used to detect fine-grained (five classes) of emotion valence.

- We designed and validated seven different gesticulations for the robot that express increasing levels of positivity.

- We demonstrated a dynamical system to compensate for varying utterance frequency and prediction errors coming from the emotion recognition component.

- We extended the framework so that utterance analysis is escalated through multiple levels in order to improve semantic interpretation of the speaker's utterance across multiple domains.

9.1.2 Bilingual Extensions

A companion robot that could understand and speak multiple languages would be welcome in communities where the caregiver and robot guardian are bilingual. The following investigations contributed extending the framework to add a multilingual capability.

- We investigated and created a computational model of a top-down language control based on the Inhibitory Control theory and validated the theory's predictions.
- We created a computational model for a psychological theory of bilingual memory in which the model parameters are learned from a large multilingual corpus. This forms the basis of studying whether bilingual effects such as cross-language interference are important for the naturalistic functioning of the companion robot, e.g., when code-switching.
- We demonstrated the extensions to the framework to incorporate bilingualism.

9.1.3 Modeling Human Language Performance

We conducted two investigations which demonstrate that it is possible to build a biologically plausible model that can fit human performance.

- We designed a neural field model that replicated the effect of immediate auditory repetition of monosyllabic words and fits it to a component of a well-studied mechanism for analyzing language processing, the event-related potential (ERP).
- We connected to neural fields and demonstrated the cognitive process of word recognition by created a neural field model of Marslen-Wilson's Cohort theory.
- We showed how a neural field component can be connected to the framework in such a way that in could allow the robot to monitor and adjust task performance.

9.2 Implications of Contributions

The contributions of this dissertation are related to each component of the situated emotion expression framework. We discuss the implications of each component as follows.

The LVASR is the first component and processes all of the information contained in the PD person's speech signal. The LVASR component, at present, extracts all acoustic features which it uses to predict words. However, there is much more information that could potentially be extracted using readily available open-source software tool-kits which can be used, for example, to detect the vocal characteristics of anyone who is withing range of the agent's microphone(s) and it might do so for whatever is being said. Although, this raises similar privacy issues as with conversational agents such as the Amazon Alexa or Google Home smart speakers, embedding such a system in a companion robot raises additional issues. The human may make assumptions about what the robot has heard and how it should react and if it fails to meet the human's expectations, they may become less engages or more depressed. These concerns needs to be evaluated in a controlled HRI experiment.

Similarly, the emotion detection component uses the text of the utterances from the LVASR to make inferences about the emotional state of anyone near the microphones. While the LVASR is generally accurate, it nonetheless performs at approximately a 13% WER, which is considered near state-of-the-art. This means that one in every eight words will not be recognized properly, on average. These errors propagate through the system and left unchecked can results in improper emotion inference by the detector and incorrect semantic interpretation by the NLU components. The emotion expressor and the multi-level processing of the framework attempt to minimize and some cases re-interpret the utterance to minimize such errors, but they inevitably will occur. While the HRI experiments described in Chapter 5 does explore how well the robot responded matched with what was being said, it was not intended to be a systematic exploration of the effect of errors on either the observer or the PD person. However, the implication, is that more generally a Spoken Dialog System framework, should attempt to mitigate errors arising from the start of the pipeline at the ASR and propagating through interpretation and the agent's response.

Our preliminary work on bilingualism was motivated by the implication of a relatively large community of bilinguals both within the U.S. and globally. A social robot that does not incorporate a multilingual capability will, over time, be limited to situations in which the person is only monolingual, or if the person is bilingual, may perhaps not be quite as engaging. These implications are dependent upon systematic human-robot interaction studies.

The motivation behind our investigation into neural fields was to explore what could be learned from a biologically-inspired model of human language processing. In our study, we found that the neural field can correlate with human physiological measures of performance. The implication is that it is possible to study intelligent agents that replicate, rather than exceed human performance. This is useful in studying: the cognitive processes underlying word processing, how to optimize language comprehension in demanding scenarios, and human error performance. In a companion robot, for example, this implies creating an agent which understands human performance and can adapt its behavior accordingly, perhaps anticipating and mitigating human error. The field might also regulate how fallible behavior is expressed in the agent's responses to make is more human-like and engaging.

9.3 Future Work

We explained in the Introduction (Chapter 1), that this dissertation is part of a larger project to develop a socially assistive robot for the self-management of health of people living with PD. At a minimum, the items listed below would be necessary before clinical trials with people with PD and their caregivers could begin.

- Detect continuous values of emotion and arousal.

- Add the capability to observe, detect, and mediate emotion among several persons.

- Investigate other emotion detection approaches, e.g., deep learning language models.
- Evaluate whether the expressive companion robot changes an observer's opinion of the PD person's emotional characteristics.
- Compare efficacy of each expressive channel, individually and together: e.g., facial (via virtual agent), gesticulation, vocal.
- Investigate combining prosodic features as well as language features in the audio signal for emotion detection.

The objective of developing the framework is more than to simply design a more engaging robot; it is to develop the underlying components that would enable a person living with Parkinson's disease to manage their self-care. More generally, the framework's capacity to detect emotion in conversational speech or text is usable in a variety of situations in which it is important to infer another person's mental state. This might occur in noisy communications channels when facial expressions or vocal tone may not be available, or in fast moving, urgent situations: e.g., in command and control battlefield situations or for air-traffic control. This might occur when new team members are added to a remote collaboration environment and when it is not possible to detect social cues from video or audio channel. Finally, extending the dynamical system component and emotion detector to monitor several conversations could allow an mediator, for example, monitor and help regulate the emotional content in the exchange among the parties.

9.4 A Final Word

I end this dissertation emphasizing Trust. It is an exciting time to be a researcher in computer science, cognitive science, and human-robot interaction. Speech, vision, and language processing have advanced over the last 10 to 15 years that have made possible non-situated agents such as smartphones, conversational agents, and near-self-driving cars. Situated agents, robots, present a larger challenge that is both technical and human. While we believe many of the technical challenges (e.g., human-like locomotion, one-shot learning) will eventually be overcome, the human-robot challenge is more difficult. Humans tend to anthropomorphize many items such as dolls, iRobot Roombas, and the degree to which they trust these agents is an area under investigation. The consequence when the intelligent agent breaks that trust is not fully known. Thus, as we build companion robots for a vulnerable population, they should be designed with this bonding in mind. One consequence is that the robot's behavior should be predictable, as we expect of our human companions. If the robot is to infer the human's mental state, then it should be resilient to prediction errors as much as possible. If the robot is the companion of a person who speaks more than one language, then it too should have that capability. If human performance varies, then the robot should be able to infer this variance and adjust its behavior accordingly.

If the robot is unpredictable and its performance in not what the human expected, it can lower trust and, consequently, engagement levels. It is my goal that this work not only contributes to creating an emoting robot companion to help a vulnerable population manage their self-care, but that it also sparks further research into the inevitable bond humans are likely to form with their silicon companions and that this is used to inform the robot's design.

Bibliography

Neurological disorders: Public health challenges. https://www.who.int/mental_health/neurology/neurodiso/en/, 2006. Accessed: 2019-03-28.

ASpIRE IARPA automatic speech recognition in reverberant environments challenge. https://www.iarpa.gov/index.php/newsroom/iarpa-in-the-news/2014/410-aspire-iarpa-automatic-speech-recognition-in-reverberant-environments-challenge, 2015a. Accessed: 2019-11-12.

Four teams win IARPA's ASpIRE challenge. https://www.afcea.org/content/?q=Blog-four-teams-win-iarpas-aspire-challenge, 2015b. Accessed: 2019-11-12.

Microsoft translator. https://www.microsoft.com/en-us/translator/, 2018. Accessed: 2018-12-15.

Parkinson's disease: Challenges, progress, and promise. https://www.ninds.nih.gov/Disorders/All-Disorders/Parkinsons-Disease-Challenges-Progress-and-Promise, 2018. Accessed: 2018-10-29.

Ibm tone analyzer: The science behind the service. https://cloud.ibm.com/docs/services/tone-analyzer?topic=tone-analyzer-ssbts#the-science-behind-the-service, 2019. Accessed: 2019-12-12.

Oxford english dictionary. https://www.oed.com/view/Entry/61249?rskey=M3aulB&result=1#eid, 2020. Accessed: 2020-01-22.

Merriam-webster online dictionary. `https://www.merriam-webster.com`, 2020. Accessed: 2020-01-19.

Pepper user guide. `http://doc.aldebaran.com/2-4/family/pepper_user_guide/setting_menu_advanced_pep.html`, 2020. Accessed: 2020-2-18.

Ameeta Agrawal and Aijun An. Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 346–353. IEEE Computer Society, 2012.

Sagar Ahire. A survey of sentiment lexicons, 2014.

S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological cybernetics*, 27(2):77–87, 1977.

Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. *Proc. Graphics Interface Conf*, 12 2000.

Akshay Amolik, Niketan Jivane, Mahavir Bhandari, and M Venkatesan. Twitter sentiment analysis of movie reviews using machine learning techniques. *international Journal of Engineering and Technology*, 7(6):1–7, 2016.

Alexandra Balahur. Sentiment analysis in social media texts. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 120–128, 2013.

Alexandra Balahur, Jesus M Hermida, and Andres Montoyo. Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. *IEEE transactions on affective computing*, 3(1):88–101, 2011.

David A Balota, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman. The english lexicon project. *Behavior research methods*, 39(3):445–459, 2007.

Etienne Barnard, Marelie H Davel, and Gerhard B Van Huyssteen. Speech technology for information access: a south african case study. In *2010 AAAI Spring Symposium Series*, 2010.

R. D. Beer. Dynamical approaches to cognitive science. *Trends in Cognitive Science*, 4(3):91–99, 2000.

Maxwell Bennett, Tom Williams, Daria Thames, and Matthias Scheutz. Differences in interaction patterns and perception for teleoperated and autonomous humanoid robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6589–6594. IEEE, 2017.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100, 2014.

E. Blanco-Elorrieta and L. Pylkkänen. Bilingual language switching in the laboratory versus in the wild: The spatiotemporal dynamics of adaptive language control. *The Journal of Neuroscience*, 37(37):9022–9036, 2017.

D. M. Blei, A. Y. Eng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Language Research*, 3:993–1022, 2003a.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003b.

Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology . . . , 1999.

M.C. Brady. A bi-directional graphical model for babble-feedback learning in speech. *Procedia Computer Science*, 41:220–225, 2014.

Michael C. Brady. *A field-based artificial neural network with cerebellar model for complex motor sequence learning.* PhD thesis, Indiana University, 2012.

C. Burgess and K. Lund. Modeling parsing constraints with high dimensional context space. *Language and Cognitive Processes*, 12:177–210, 1997.

Kristin Byron. Carrying too heavy a load? the communication and miscommunication of emotion by email, 2008.

Yitao Cai and Xiaojun Wan. Multi-domain sentiment classification based on domain-aware embedding and attention. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4904–4910. AAAI Press, 2019.

R. A. Calvo and S. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1 (1):18–37, Jan 2010. ISSN 1949-3045. doi: 10.1109/T-AFFC.2010.1.

Erik Cambria and Amir Hussain. *Sentic computing: Techniques, tools, and applications*, volume 2. Springer Science & Business Media, 2012.

Erik Cambria, Catherine Havasi, and Amir Hussain. Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *Twenty-fifth international FLAIRS conference*, 2012a.

Erik Cambria, Andrew Livingstone, and Amir Hussain. The hourglass of emotions. In *Cognitive behavioural systems*, pages 144–157. Springer, 2012b.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Pin-Jung Chen, I-Hung Hsu, Yi-Yao Huang, and Hung-Yi Lee. Mitigating the impact of speech recognition errors on chatbot using sequence-to-sequence model. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 497–503. IEEE, 2017.

Wei Chen, Sankaranarayanan Ananthakrishnan, Rohit Kumar, Rohit Prasad, and Prem Natarajan. Asr error detection in a conversational spoken language translation system. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7418–7422. IEEE, 2013.

J. Chuang, J. Wu, R. Socher, R. Ravisundaram, and T. Tayyab. Stanford sentiment treebank. `https://nlp.stanford.edu/sentiment/`, 2013. Accessed: 2020-3-19.

A. Collins and E. Loftus. A spreading activation theory of semantic processing. *Psychological Review*, 82:407–428, 1975.

David R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958.

Henriette Cramer, Jorrit Goddijn, Bob Wielinga, and Vanessa Evers. Effects of (in) accurate empathy and situational valence on attitudes towards robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 141–142. IEEE, 2010.

George S Cree, Ken McRae, and Chris McNorgan. An attractor model of lexical conceptual processing: Simulating semantic priming. *Cognitive Science*, 23(3): 371–414, 1999.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondrej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. *Alexa Prize Proceedings*, 2018.

Delphine Dahan and James S. Magnuson. Spoken Word Recognition. In Matthew J. Traxler and Morton A. Gemsbacher, editors, *Handbook of Psycholinguistics*. 2006.

M. Dambacher, R. Kliegl, M. Hofmann, and A.M. Jacobs. Frequency and predictability effects on event-related potentials during reading. *Brain Research*, 1084(1):89–103, 2006.

Charles Darwin and Phillip Prodger. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

Mark Davies. Word frequency data, 2018. https://www.wordfrequency.info/spanish.asp, last accessed on 2020-02-16.

B. de Gelder, A.W. de Borst, and R. Watson. The perception of emotion in body expressions. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(2):149–158, 2015. doi: 10.1002/wcs.1335. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcs.1335.

P. Ravindra De Silva and Nadia Bianchi-Berthouze. Modeling human affective postures: an information theoretic characterization of posture features. *Computer Animation and Virtual Worlds*, 15(3-4):269–276, 2004. doi: 10.1002/cav.29. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cav.29.

Scott Deerwester, Susan Dumais, Thomas Landauer, George Furnass, and Laura Beck. Improving information retrieval with latest semantic indexing. *In: ASIS '88. Information Technology: Planning for the next fifty years. Proceedings of the First Annual Meeting of the American Society for Information Science, Volume 25, Atlanta, Georgia, 23-27 October 1988 Edited by Christine L. Borgman and Edward Y. H. Pai*, 10 1988.

E. DeGroat, K. D. Lyons, and L. Tickle-Degnen. Verbal content during favorite activity interview as a window into the identity of people with Parkinson's disease. *Occupational Therapy Journal of Research: Occupation, Participation, and Health*, 26(2), 2006.

Margaret Deuchar, Peredur Davies, Jon Russell Herring, M Parafita Couto, and Diana Carter. Building bilingual corpora. 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

T. Dijkstra and W. Van Heuven. The architecture of the bilingual word recogniton system: From identification to decision. *Bilingualism: Language and Cognition*, 5(3):175–179, 2002.

Ton Dijkstra, Alexander Wahl, Franka Buytenhuijs, Nino Van Halem, Zina Al-Jibouri, Marcel De Korte, and Steven Rekké. Multilink: a computational model for bilingual word recognition and word translation. *Bilingualism: Language and Cognition*, 22(4):657–679, 2019.

Joan DiMicco and David Millen. Identity management: Multiple presentations of self in facebook. pages 383–386, 01 2007. doi: 10.1145/1316624.1316682.

Y. Dong, S. Gui, and B. MacWhinney. Shared and separate meanings in the bilingual mental lexicon. *Bilingualism: Language and Cognition*, 8(3):221–238, 2005.

Sri Harsha Dumpala, Imran Sheikh, Rupayan Chakraborty, and Sunil Kumar Kopparapu. Sentiment classification on erroneous asr transcripts: A multi view learning approach. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 807–814. IEEE, 2018.

Derek Edwards. Emotion discourse. *Culture & psychology*, 5(3):271–291, 1999.

Paul Ekman. Expression and the nature of emotion. *Approaches to emotion*, 3: 19–344, 1984.

Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992. doi: 10.1080/02699939208411068. URL https://doi.org/10.1080/02699939208411068.

Paul Ekman and Daniel Cordaro. What is meant by calling emotions basic. *Emotion review*, 3(4):364–370, 2011.

J.L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer, 2006.

Mathias Etcheverry and Dina Wonsever. Spanish word vectors from wikipedia. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3681–3685, 2016.

Sarah Fdili Alaoui, Cyrille Henry, and Christian Jacquemin. Physical modelling for interactive installations and the performing arts. *International Journal of Performance Arts and Digital Media*, 10(2):159–178, 2014.

Beverley Fehr and James A Russell. Concept of emotion viewed from a prototype perspective. *Journal of experimental psychology: General*, 113(3):464, 1984.

Jamie Fraser, Ioannis Papaioannou, and Oliver Lemon. Spoken conversational ai in video games: Emotional dialogue management increases user engagement. In *IVA*, pages 179–184, 2018.

K. Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society London, Series B, Biological Sciences*, 360(1456):815–836, 2005.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David S Pallet. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM, NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn & TensorFlow.* O'Reilly, Sebastopol, CA, 1 edition, 2017.

Felix Gervits, Dean Thurston, Ravenna Thielstrom, Terry Fong, Quinn Pham, and Matthias Scheutz. Toward genuine robot teammates: Improving human-robot team performance using robot shared mental models. In *AAMAS '20: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 2020.

Arnab Ghoshal, Pawel Swietojanski, and Steve Renals. Multilingual training of deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7319–7323. IEEE, 2013.

Christopher G Goetz, Werner Poewe, Olivier Rascol, Cristina Sampaio, Glenn T Stebbins, Carl Counsell, Nir Giladi, Robert G Holloway, Charity G Moore, Gregor K Wenning, et al. Movement disorder society task force report on the hoehn and yahr staging scale: status and recommendations the movement disorder society task force on rating scales for parkinson's disease. *Movement disorders*, 19 (9):1020–1028, 2004.

D. W. Green. Mental control of the bilingual lexico-semantic system. *Bilingualism: Language and Cognition*, 1:67–81, 1998.

D.W. Green and J. Abutalebi. Language control in bilinguals: The adaptive control hypothesis. *Journal of Cognitive Psychology*, 25:515–530, 2013.

F. Grosjean. Processing mixed languages: Issues, findings, and models. In Li Wei, editor, *The Bilingualism Reader*. Routledge, New York, NY, 1997.

James J Gross and Ross A Thompson. Emotion regulation: Conceptual foundations. In J. J. Gross, editor, *Handbook of emotion regulation*. Guilford Press, 2007.

S. Grossberg. Adaptive resonance theory. In L. Nadel, editor, *The Encyclopedia of Cognitive Science*. Wiley, 1 edition, 2005.

Taabish Gulzar, Anand Singh, Dinesh Kumar Rajoriya, and Najma Farooq. A systematic analysis of automatic speech recognition: an overview. *Int. J. Curr. Eng. Technol*, 4(3):1664–1675, 2014.

William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access, 2016.

M. Han, C. Lin, and K. Song. Robotic emotional expression generation based on mood transition and personality model. *IEEE Transactions on Cybernetics*, 43 (4):1290–1303, Aug 2013. ISSN 2168-2267. doi: 10.1109/TSMCB.2012.2228851.

Eddie Harmon-Jones, Cindy Harmon-Jones, David M Amodio, and Philip A Gable. Attitudes toward emotions. *Journal of personality and social psychology*, 101(6): 1332, 2011.

M. Harper. The automatic speech recogition in reverberant environments (aspire) challenge. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 547–554, Dec 2015. doi: 10.1109/ASRU.2015.7404843.

Mary Harper. The automatic speech recogition in reverberant environments (aspire) challenge. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 547–554. IEEE, 2015.

S.A. Hillyard and Lourdes Anllo-Vento. Event-related brain potentials in the study of visual selective attention. *Proceedings of the National Academy of Sciences of the United States of America*, 95(3):781–787, 1998.

Elliott M Hoey and Kobin H Kendrick. Conversation analysis. *Research methods in psycholinguistics: A practical guide*, pages 151–173, 2017.

Hartwig Holzapfel. Towards development of multilingual spoken dialogue systems. In *Proceedings of the 2nd Language and Technology Conference*, 2005.

J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, volume 79, pages 2554–2558, 1982.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

C. Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *International AAAI Conference on Web and Social Media*, 2014.

William James. Discussion: The physical basis of emotion. *Psychological review*, 1 (5):516, 1894.

N. F. Johnson and K. R. Pugh. A cohort model of visual word recognition. *Cognitive Psychology*, 26:240–346, 1994.

Kristiina Jokinen and Michael McTear. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies*, 2(1):1–151, 2009.

Dan Jurafsky. *Speech & language processing.* Pearson Education India, 2000.

Patrik N Juslin and Klaus R Scherer. Vocal expression of affect. *The new handbook of methods in nonverbal behavior research*, pages 65–135, 2005.

Sunghwan Mac Kim, Alessandro Valitutti, and Rafael A Calvo. Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 62–70. Association for Computational Linguistics, 2010.

Svetlana Kiritchenko and Saif M Mohammad. The effect of negators, modals, and degree adverbs on sentiment composition. *arXiv preprint arXiv:1712.01794*, 2017.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

Dave F. Kleinschmidt and T. Florian Jaeger. Robust speech perception: Recognize the familiar, generalize to the similar, adapt to the novel. *Psychologicial Review*, 122(2):148–203, 2015.

Christof Koch. *The Quest for Consciousness: a Neurobiological Approach.* Roberts and Company Publishers, Englewood, CO, 1 edition, 2004.

Robert M Krauss, William Apple, Nancy Morency, Charlotte Wenzel, and Ward Winton. Verbal, vocal, and visible factors in judgments of another's affect. *Journal of Personality and Social Psychology*, 40(2):312, 1981.

J.F. Kroll and E. Stewart. Category interference in translation and picture naming: Evidence for asymmetric connections between bilingual memory representations. *Journal of Memory and Language*, 33:149–174, 1994.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Gina R. Kuperberg. The proactive comprehender: What event-related potentials tell us about the dynamics of reading comprension. In B. Miller, L. Cutting, and P. McCardle, editors, *Unraveling the Behavioral, Neurobiological, and Genetic Components of Reading Comprehension*, pages 176–192. Paul Brookes Publishing, 2013.

Gina R. Kuperberg. Separate streams or probabilistic inference? What the N400 can tell us about the comprehension of events. *Language, Cognition and Neuroscience*, 2016. doi: 10.1080/23273798.2015.1130233.

Gina R. Kuperberg and T. Florian Jaeger. What do we mean by prediction in language comprehension? *Language, Cognition and Neuroscience*, 2015. doi: 10.1080/23273798.2015.1102299.

Ian Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. Out-of-domain utterance detection using classification confidences of multiple topics. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):150–161, 2006.

Ian R Lane, Tatsuya Kawahara, Tomoko Matsui, and Satoshi Nakamura. Dialogue speech recognition by combining hierarchical topic classification and language model switching. *IEICE transactions on information and systems*, 88(3):446–454, 2005.

S. Laszlo and D.C. Plaut. A neurally plausible parallel distributed processing model of event-related potential reading data. *Brain and Language*, 120:271–281, 2012a.

Sarah Laszlo and David C Plaut. A neurally plausible parallel distributed processing model of event-related potential word reading data. *Brain and language*, 120(3): 271–281, 2012b.

Niklas Laxström, Kristiina Jokinen, and Graham Wilcock. Situated interaction in a multilingual spoken information access framework. In *Situated Dialog in Speech-Based Human-Computer Interaction*, pages 165–175. Springer, 2016.

Richard S Lazarus and Richard S Lazarus. *Emotion and adaptation.* Oxford University Press on Demand, 1991.

Chul Min Lee, Shrikanth S Narayanan, et al. Toward detecting emotions in spoken dialogs. *IEEE transactions on speech and audio processing*, 13(2):293–303, 2005.

Anton Leuski and David Traum. Practical language processing for virtual humans. In *Twenty-Second IAAI Conference*, 2010.

W.J.M. Levelt and A.S. Meyer. A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22:1–75, 1999.

P. Li and I. Farkas. A self-organizing connectionist model of bilingual processing. In R. Heredia and J. Altarriba, editors, *Bilingual Sentence Processing*, volume 134. North Holland, 2002.

Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

Donald B Lindsley. Emotion. 1951.

B. Liu. Sentiment analysis and subjectivity. In N. Indurkhya and F. Damerau, editors, *Handbook of Natural Language Processing*. Chapman and Hall, San Rafael, CA, 2 edition, 2010.

S. J. Luck. *An Introduction to the Event-related Potential Technique*. The MIT Press, Cambridge, MA, 2 edition, 2014.

H. Ma, M. Saint-Hilaire, C. A. Thomas, and L. Tickle-Degnen. Stigma as a key determinant of health-related quality of life in Parkinson's disease. *Quality of Life Research*, 25(2):3037–3045, 2016.

J. Macnamara and S.L. Kushnir. Linguistic independence of bilinguals: The input switch. *Journal of Verbal Learning and Verbal Behaviour*, 10:480–487, 1971.

Suresh Manandhar, Ion Androutsopoulos, Dimitris Galanis, Harris Papageorgiou, John Pavlopoulos, and Maria Pontiki. Semeval-2014 Task 4. `http://alt.qcri.org/semeval2014/task4/`, 2014. Accessed: 2020-3-17.

William D. Marslen-Wilson. Functional parallelism in spoken word-recognition. *Cognition*, 25:71–102, 1987.

M.P Mattson. Superior pattern processing is the essence of the evolved human brain. *Frontiers in Neuroscience*, 8(265), 2014. doi: 10.3389/fnins.2014.00265.

Mirko Mazzoleni, Gabriele Maroni, and Fabio Previdi. Unsupervised learning of fundamental emotional states via word embeddings. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE, 2017.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: Part 1 an account of basic findings. *Psychological Review*, 88:375–407, 1981a.

James L. McClelland and David E. Rumelhart. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises.* The MIT Press, Cambridge, MA, 1989.

J.L. McClelland and J.L. Elman. The TRACE model of speech perception. *Cognitive Psychology*, 18(1):1–86, 1986.

J.L. McClelland and D. Rumelhart. An Interactive Activation Model of Contextual Effects in letter perception: Part 1. *Psychology Review*, 88(5):375–407, 1981b.

Derek McColl and Goldie Nejat. Recognizing emotional body language displayed by a human-like social robot. *International Journal of Social Robotics*, 6, 04 2014. doi: 10.1007/s12369-013-0226-7.

Michael F McTear. Modelling spoken dialogues with state transition diagrams: experiences with the cslu toolkit. In *Fifth International Conference on Spoken Language Processing*, 1998.

Albert Mehrabian. *Nonverbal communication.* Transaction Publishers, 1972.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Saif M Mohammad. Challenges in sentiment analysis. In *A practical guide to sentiment analysis*, pages 61–83. Springer, 2017.

D.L. Molfese, A. P. Fonaryova Key, M. J. Maguire, G.O. Dove, and V.J. Molfese. Event-related potentials (ERPs) in speech perception. In D.B. Pisoni and R.E. Remez, editors, *The Handbook of Speech Perception.* Blackwell Publishing, 2005.

Renato De Mori. *Spoken dialogues with computers.* Academic Press, Inc., 1997.

Atreyee Mukherjee, Sandra Kübler, and Matthias Scheutz. Creating pos tagging and dependency parsing experts via topic modeling. In *Proceedings of the 15th*

Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 347–355, 2017.

Iain R Murray and John L Arnott. Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *The Journal of the Acoustical Society of America*, 93(2):1097–1108, 1993.

Carol Myers-Scotton. Multiple voices: An introduction to bilingualism. 2006.

Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

D. A. Norman and T. Shallice. Attention to action: Willed and automatic control of behavior. In R. J. Davidson, G. E. Schwartz, and D. Shapiro, editors, *Consciousness and Self-regulation*, volume 4. Plenum Press, New York, NY, 1986.

D. Norris. A dynamic-net model of human speech recognition. In G.T.M. Altmann, editor, *Cognitive Models of Speech Processing.* MIT Press, Cambridge, MA, 1995.

Kyo-Joong Oh, DongKun Lee, Chanyong Park, Young-Seob Jeong, Sawook Hong, Sungtae Kwon, and Ho-Jin Choi. Out-of-domain detection method based on sentence distance for dialogue systems. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 673–676. IEEE, 2018.

R.C. Oldfield. The assessment and analysis of handedness: the edinburgh inventory. *Neuropsychologia*, 9(1):97–113, 1971.

Oxford University Press. Dictionary facts. `https://www.oed.com/page/facts/loginpage`, 2020. Accessed: 2020-3-19.

Bo Pang and Lillian Lee. Movie review data. `http://www.cs.cornell.edu/people/pabo/movie-review-data/`, 2012. Accessed: 2020-3-19.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-*

02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118704. URL `https://doi.org/10.3115/1118693.1118704`.

Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

S.-T Park, Lilia Moshkina, and Ronald Arkin. Recognizing nonverbal affective behavior in humanoid robots. *Intelligent Autonomous Systems 11, IAS 2010*, 01 2010. doi: 10.3233/978-1-60750-613-3-12.

Aneta Pavlenko. Conceptual representation in the bilingual lexicon and second language vocabulary learning. *The bilingual mental lexicon: Interdisciplinary approaches*, pages 125–160, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. W. Pennebaker and M. E. Francis. Cognitive, emotional, and language processes in disclosure. *Cognition and Emotion*, 10:601–626, 1996.

J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. *The development and psychometric properties of LIWC2015.* University of Texas at Austin, Austin, TX, 2015a. doi: 10.15781/T29G6Z.

J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. *The development and psychometric properties of LIWC2015.* University of Texas at Austin, Austin, TX, 2015b. doi: 10.15781/T29G6Z.

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)*, pages 1532–1543, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Rosalind W Picard. Affective computing-mit media laboratory perceptual computing section technical report no. 321. *Cambridge, MA*, 2139, 1995.

Rosalind W Picard. *Affective computing*. MIT press, 2000.

Robert Plutchik. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier, 1980.

Robert Plutchik. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219, 1984.

Robert Plutchik and Henry Kellerman. *Theories of emotion*, volume 1. Academic Press, 2013.

Soujanya Poria, Erik Cambria, Grégoire Winterstein, and Guang-Bin Huang. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69:45–63, 2014.

Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

A. Protopapas. Connectionist modeling of speech perception. *Psychological Bulletin*, 125(4):410–436, 1999.

M. Rabovsky and K. McRae. Simulating the N400 erp component as semantic network error: insights from a feature-based connectionist attractor model of word meaning. *Cognition*, 132:68–89, 2014a.

Milena Rabovsky and Ken McRae. Simulating the n400 erp component as semantic network error: Insights from a feature-based connectionist attractor model of word meaning. *Cognition*, 132(1):68–89, 2014b.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*, 2018.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

Ranier Reisenzein. Pleasure-arousal theory and the intensity of emotions. *Journal of personality and social psychology*, 67(3):525, 1994.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3317-7.

M.D. Rugg. Dissociation of semantic priming, word and non-word repetition effects by event-related potentials. *The Quarterly Journal of Experimental Psychology*, 39(1):123–148, 1987.

David Rumelhart and P. Siple. Process of recognizing tachistoscopically presented words. *Psychology Review*, 81:99–118, 1974.

James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

Sepideh Sadeghi, He Pu, Matthias Scheitz, Phillip Holcomb, and Katherine Midgley. A pdp model for capturing n400 effects in early l2 learners during bilingual word reading tasks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35, 2013.

Arup Sarma and David D Palmer. Context-based speech recognition error detection and correction. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 85–88. Association for Computational Linguistics, 2004.

Klaus R Scherer. Adding the affective dimension: a new look in speech analysis and synthesis. In *ICSLP*, 1996.

Matthias Scheutz, Evan Krause, Brad Oosterveld, Tyler Frasca, and Robert Platt. Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 2017.

Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. In *Cognitive Architectures*, pages 165–193. Springer, 2019a.

Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection diarc architecture. In *Cognitive Architectures*, pages 165–193. Springer, 2019b.

G. Schöner and JA Kelso. Dynamic pattern generation in behavioral neural systems. *Science*, 239(4847):1513–1520, 1988.

G. Schöner and J. Spencer. *Dynamic Thinking: a primer on dynamic field theory.* Oxford University Press, New York, NY, 2015.

Björn Schuller, Raquel Jiménez Villar, Gerhard Rigoll, and Manfred Lang. Meta-classifiers in acoustic and linguistic feature fusion-based affect recognition. In

Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., volume 1, pages I–325. IEEE, 2005.

M. Shah, L. Miao, C. Chakrabarti, and A. Spanias. A speech emotion recognition framework based on latent dirichlet allocation: Algorithm and fpga implementation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2553–2557, May 2013. doi: 10.1109/ICASSP.2013.6638116.

Mohit Shah, Chaitali Chakrabarti, and Andreas Spanias. Within and cross-corpus speech emotion recognition using latent topic model-based features. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):4, Jan 2015. ISSN 1687-4722. doi: 10.1186/s13636-014-0049-y. URL `https://doi.org/10.1186/s13636-014-0049-y`.

C. Shan, S. Gong, and P. W. McOwan. Beyond facial expressions: Learning human emotion from body gestures. In *Proc. BMVC*, pages 43.1–43.10, 2007. ISBN 1-901725-34-0. doi: 10.5244/C.21.43.

Imran Sheikh, Dominique Fohr, Irina Illina, and Georges Linares. Modelling semantic context of oov words in large vocabulary continuous speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(3):598–610, 2017.

Craig A Smith, Richard S Lazarus, et al. Emotion and adaptation. *Handbook of personality: Theory and research*, pages 609–637, 1990.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560. ACM, 2008.

I. H. Sturkenboom, M. J. Graff, G. F. Borm, E. M. Adang, M. W. Nijhuis-van der Sanden, B. R. Bloem, and M Munneke. Effectiveness of occupational therapy in Parkinson's disease: study protocol for a randomized controlled trial. *Trials*, 14 (34), 2013. doi: 10.1186/1745-6215-14-34.

Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.

Ron Sun. *The Cambridge handbook of computational psychology*. Cambridge University Press, 2008.

Kiritchenko Svetlana, Zhu Xiaodan, and MM Saif. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

K. Takahashi, L. Tickle-Degnen, W. J. Coster, and N. K. Latham. Expressive behavior in Parkinson's disease as a function of interview context. *American Journal of Occupational Therapy*, 64(3):484–495, 2010.

Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010a. doi: 10.1177/0261927X09351676.

Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010b.

The Apache Software Foundation. Ordinary differential equations integration. `https://commons.apache.org/proper/commons-math/userguide/ode.html`, 2019. Accessed: 2019-08-15.

L. Tickle-Degnen, J. Hall, and R. Rosenthal. Nonverbal behavior. *Encyclopedia of Human Behavior*, 3:293–302, 1994.

L. Tickle-Degnen, T.D. Ellis, M. Saint-Hilaire, C. Thomas, and R. C. Wagenaar. Self-management rehabilitation and health-related quality of life in Parkinson's disease: A randomized controlled trial. *Movement Disorders*, 25:194–204, 2010.

Linda Tickle-Degnen and Kathleen Doyle Lyons. Practitioners' impressions of patients with parkinson's disease: the social ecology of the expressive mask. *Social Science & Medicine*, 58(3):603–614, 2004.

Gokhan Tur, Anoop Deoras, and Dilek Hakkani-Tür. Detecting out-of-domain utterances addressed to a virtual personal assistant. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

A. Valenti, B. Oosterveld, and M. Scheutz. A neural field model of word recognition. In I Juvina, Houpt J., and C. Myers, editors, *Proceedings of the 16th International Conference on Cognitive Modeling*, pages 194–199, Madison, WI: University of Wisconsin, 2018.

Andrew P. Valenti and Matthias J. Scheutz. A computational model of bilingual inhibitory control in a lexical decision task. In *The 12th International Conference on Cognitive Modeling*, 2013. URL `http://iccm-conference.org/2013-proceedings/papers/0042/index.html`.

Andrew P. Valenti, Michael C. Brady, Matthias J. Scheutz, Phillip J. Holcomb, and He Pu. A neural field model of word repetition effects in early time-course ERPs in spoken word perception. In A.Papafragou, D. Grodner, D. Mirman, and J.C. Trueswell, editors, *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, pages 2765–2770, 2016. ISBN 978-0-9911967-3-9.

Andrew P. Valenti, Meia Chita-Tegmark, Michael Gold, Theresa Law, and Matthias

Scheutz. In their own words: A companion robot for detecting the emotional state of persons with Parkinson's disease. In *11th International Conference on Social Robotics*, pages 1–10, Madrid, Spain, November 2019a. Springer.

Andrew P. Valenti, Meia Chita-Tegmark, Theresa Law, Alexander W. Bock, Bradley Oosterveld, and Matthias Scheutz. When your face and tone of voice don't say it all: Inferring emotional state from word semantics and conversational topics. In *Workshop on Cognitive Architectures for HRI: Embodied Models of Situated Natural Language Interactions at AAMAS 2019*, Montreal, Canada, May 2019b.

Andrew P. Valenti, Meia Chita-Tegmark, Linda Tickle-Degnen, Alexander W. Bock, and Matthias J. Scheutz. Using topic modeling to infer the emotional state of people living with Parkinson's disease. *Assistive Technology*, pages 1–10, 2019c. URL `https://doi.org/10.1080/10400435.2019.1623342`.

Andrew P. Valenti, Avram Bock, Meia Chita-Tegmark, Michael Gold, and Matthias Scheutz. Emotion expression in a socially assistive robot for persons with Parkinson's disease. In *13th PErvasive Technologies Related to Assistive Environments Conference (PETRA '20)*, Corfu, Greece, July 2020a. ACM.

Andrew P. Valenti, Ravenna Thielstrom, Felix Gervits, Michael Gold, and Matthias Scheutz. A multi-level framework for understanding spoken dialog using topic detection. In *Submitted to: The 21st Annual SIGdial Meeting on Discourse and Dialogue*, Boise, Idaho, July 2020b. ACL.

Charl Van Heerden, Etienne Barnard, and Marelie Davel. Basic speech recognition for spoken dialogues. 2009.

W. J. B. Van Heuven, A. (Ton) Dijkstra, and J. Grainger. Orthographic neighborhood effects in bilingual word recognition. *Journal of Memory and Language*, 39: 458–483, 1998.

C. van Petten and M. Kutas. Electrophysiological evidence for the flexibility of lexical processing. In G.B. Simpson, editor, *Understanding Word and Sentence*. North-Holland Press, Amsterdam, 1991.

C. van Petten, M. Kutas, R. Kluender, M. Mitchnier, and H. McIsaac. Fractionating the word repetition effect with event-related potentials. *Journal of Cognitive Neuroscience*, 3(2):131–150, 1991.

Krisztián Varga. Kaldi asr: Extending the aspire model, 2017. URL `https://chrisearch.wordpress.com/2017/03/11/speech-recognition-using-kaldi-extending-and-using-the-aspire-model/`.

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

Richard Veale, Gordon Briggs, and Matthias Scheutz. Linking cognitive tokens to biological signals: Dialogue context improves neural speech recognizer performance. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35, 2013.

JD Velsquez. Modeling emotions and other motivations in synthetic agents. *Aaai/iaai*, pages 10–15, 1997.

R. E. von Studnitz and D. W. Green. Lexical decision and language switching. *International Journal of Bilingualism: Cross-Linguistic Studies of Language Behaviour*, 1:3–24, 1997.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892. IEEE, 2012.

Catherine Wagongne, Jean-Pierre Changeux, and Stanislas Dehane. A theory of cortical responses. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 360(1456):815–836, 2005.

Graham Wilcock and Kristiina Jokinen. Multilingual wikitalk: Wikipedia-based talking robots that switch languages. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 162–164, 2015.

Ming Xiang and Gina R. Kuperberg. Reversing expectations during discourse comprehension. *Language, Cognition and Neuroscience*, 2014. doi: 10.1080/23273798.2014.995679.

P. Xiaolan, X. Lun, L. Xin, and W. Zhiliang. Emotional state transition model based on stimulus and personality characteristics. *China Communications*, 10(6): 146–155, June 2013. ISSN 1673-5447.

Wei Xu and Alexander Rudnicky. Task-based dialog management using an agenda. In *ANLP-NAACL 2000 Workshop: Conversational Systems*, 2000.

Guo L. Yang, Jin H. Zhang, and Hui Sun. Design of emotional interaction system based on affective computing model. *Applied Mechanics and Materials*, 198-199: 367, 09 2012. URL `https://login.ezproxy.library.tufts.edu/login?url=https://search.proquest.com/docview/1443259693?accountid=14434`. Copyright - Copyright Trans Tech Publications Ltd. Sep 2012; Last updated - 2018-10-05.

Emre Yilmaz, Henk van den Heuvel, and David A van Leeuwen. Investigating bilingual deep neural networks for automatic speech recognition of code-switching frisian speech. 2016.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5): 1160–1179, 2013.

Li Zhang, Ming Jiang, Dewan Farid, and M.A. Hossain. Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *Ex-*

pert Systems with Applications, 40(13):5160 – 5168, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2013.03.016. URL `http://www.sciencedirect.com/science/article/pii/S0957417413001668`.

Victor W Zue and James R Glass. Conversational interfaces: Advances and challenges. *Proceedings of the IEEE*, 88(8):1166–1180, 2000.