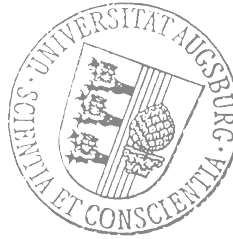


This is actually the first page of the thesis and will be discarded after the print out. This is done because the title page has to be an even page. The memoir style package used by this template makes different indentations for odd and even pages which is usually done for better readability.

University of Augsburg
Faculty of Applied Computer Science
Department of Computer Science
Bachelor's Program in Computer Science



Bachelor's Thesis

Engagement Detection

Inferring conversational engagement from nonverbal
behaviour

submitted by
Amr Abdelraouf
on 31.7.2014

Supervisor:
Prof. Dr. Elisabeth André aus Augsburg

Adviser:
MSc. Tobias Baur

Reviewers:
Prof. Dr. Elisabeth André

Abstract

Interview skills are of utmost important for a person's career and personal image. Furthermore it is an essential matter to exlude conversational engagement in an interview to give the impression of confidence and attentiveness. This thesis aims to track the engagment level of an interviewee in a mock interview situation. It tracks the verbal and nonverbal behaviour of the interviewee with respect to the ongoing context of the interview. The gathered engagement data can be further used to assess the interviewee's performance.

Statement and Declaration of Consent

Statement

Hereby I confirm that this thesis is my own work and that I have documented all sources used.

Amr Abdelraouf

Augsburg, 3.7.2014

Declaration of Consent

Herewith I agree that my thesis will be made available through the library of the Computer Science Department.

Amr Abdelraouf

Augsburg, 3.7.2014

Contents

Contents	i
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	1
1.3 Outline	1
2 Theoretical Background	3
2.1 Setup	3
2.1.1 Subject	3
2.1.2 Agent	3
3 Events	5
3.1 Event Structure	5
3.2 Sensors	5
3.2.1 Microsoft Kinect	5
3.2.2 SMI Eyetracker	6
3.2.3 Microphone	6
3.3 Scenemaker	6
3.3.1 Gaze	7
3.3.2 Speech	7
4 Main Modules	9
4.1 Mutual Facial Gaze	9
4.2 Directed Gaze	9
4.3 Adjacency Pair	9
4.4 Bachchanneling	9
5 Bayesian Network	11

6	Summery	13
A	First Appendix	17
	List of Figures	19
	List of Tables	20

Chapter 1

Introduction

1.1 Motivation

This thesis was proposed to help measure the engagement of an interviewee in a job interview situation. Through a simple mock interview the interviewee will be assessed on his/her performance. One of the most important attributes of that performance is whether or not s/he is engaged with and attentive to the interviewer. A simple playback of his/her performance coupled with the measurement of his/her engagement level will easily highlight weak spots in the mock interview.

1.2 Objectives

This thesis aims to measure the engagement levels of an interviewee through non verbal behaviour of said interview. It studies the conversational interaction with the interviewer, the responses to certain commands, his/her behaviour during certain segments of the interview.

1.3 Outline

FEL A5ER

Chapter 2

Theoretical Background

2.1 Setup

2.1.1 Subject

The subject of our experiment is the interviewee in our mock interview. As shown in figure [2.1](#) the subject is seated approximately 70 cm from a screen. The subject is asked to adjust to a number of sensors: a Microsoft Kinect camera, an SMI Eyetracker and a microphone.

2.1.2 Agent

The agent in our experiment is a reference to the interviewer. The agent is simulated by Scenemaker: a software that is used to create a virtual environment and virtual characters that follow a written script. This script contains verbal utterances, gestures, and commands to wait for a reply from the subject. The main scene used contains two virtual agents named Curtis and Gloria who are standing behind a desk to mimic an office interview. On the left lies a white board that is used as an object in our script.



Figure 2.1: Interview setup



Figure 2.2: Virtual agents

Chapter 3

Events

Events are the backbone of the software workings of this thesis. Raw sensor data are converted to events that can be further processed in the software's pipeline. Furthermore external software send events to our own software over a network. These events can be displayed by themselves as output or can be used as inputs to trigger other events.

3.1 Event Structure

Events are constructs of several attributes:

Time The clock signature of when the event was triggered.

Duration The time duration of the event.

Ptr Meta data about the event.

Type Indicates the nature of the meta data wrapped by the event.

3.2 Sensors

3.2.1 Microsoft Kinect

Kinect sensors are used to track the skeletal movements of a subject. However in this module we are mainly interested in the movement of the subject's head. Kinect is used to detect the perpetual displacement of the subject's head which indicates that s/he is nodding. This triggers an event

called *HeadNod*. HeadNod is an event measured every 500 ms and its pointer contains a value from 0 to 1 which represents the probability that the subject is nodding his/her head.

3.2.2 SMI Eyetracker

The SMI Eyetracker is used to pinpoint where the subject is currently looking. Since we are dealing with a virtual agent on a screen we consider the top left corner of the screen as the (0,0) coordinate. Displacement to the right and bottom of the looking point on the screen change the values of the x and y coordinates respectively.

The software defines two main areas on the screen. First is the area of the Agent's face. Second is the area of the board that is present in the environment.

When the subject's looking point falls on the area defined for the agent's face it triggers an event called *SubjectFacialGaze*. SubjectFacialGaze's pointer contains a value of either 0 or 1 indicating whether or not the subject is looking at the agent's face. When the gaze enters the facial area SubjectFacialGaze is triggered with the value 1 indicating that it has started and when the gaze leaves the facial area it is triggered with the value 0 indicating that it is complete.

If the subject's gaze falls in the area of the board the event *SubjectObjectGaze* is triggered. Similar to SubjectFacialGaze, the event carries a value of either 0 or 1 indicating whether or not the subject is looking at the board, value 1 when it starts and value 0 when it ends.

3.2.3 Microphone

A microphone is used to record the verbal utterances produced by the subject. When the microphone detects a voice the event *vad* (which is short for..) is fired. When the voice is first detected the event's pointer carries a value of 1. When the voice activity ends the same event is triggered but with value 0 to indicate that the event is complete.

3.3 Scenemaker

As mentioned before in subsection 2.1.2 Scenemaker is the software used to simulate the interview environment and create the virtual agents. Scenemaker is also responsible for sending the events that are triggered to rep-

resent the agents' behaviour to our software's pipeline. The events can be subcategorized into two main parts: Gaze and speech.

3.3.1 Gaze

Firstly we are concerned with where the agent is looking. When the script commands the agent to look at the subject in front of the screen the event *AgentSubjectGaze* is triggered. Similar to the subject's gaze events the event pointer holds the value 1 when the agent starts looking at the subject and holds the value 0 when the agent looks away from the subject's face. Furthermore when the agent is commanded to look at the board the event *AgentObjectGaze* is triggered with a pointer value 1 or 0 indicating that the agent has started or stopped looking at the board.

3.3.2 Speech

The agent utters the sentences that are written in the affiliated script. When the agent starts reading a sentence the event *AgentSpeech* is triggered with a pointer value of 1. When the agent finishes reading that sentence *AgentSpeech* is triggered with a pointer value of 0.

Chapter 4

Main Modules

4.1 Mutual Facial Gaze

4.2 Directed Gaze

4.3 Adjacency Pair

4.4 Bachchanneling

Chapter 5

Bayesian Network

Chapter 6

Summery

Appendices

Appendix A

First Appendix

List of Figures

2.1	Interview setup	4
2.2	Interview setup	4

List of Tables

List of Algorithms
