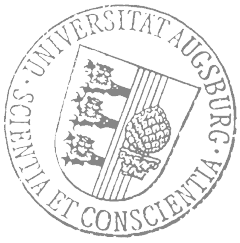
This is actually the first page of the thesis and will be discarded after the print out. This is done because the title page has to be an even page. The memoir style package used by this template makes different indentations for odd and even pages which is usally done for better readability.

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 verbak and 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 exhude 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

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 performance of the interviewee will be assessed. One of the most important attributes of that performance is whether or not the interviwee is engaged with and attentive to the interviewer. A simple playback of the interview coupled with the measurment of the engagement level will easily highlight the ups and downs of his/her demonstration in the mock interview.

1.2 Objectives

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

1.3 Outline

This thesis will first discuss the theoretical background and the information gathered on this subject. It will go into the details of the work previously done. Next it will describe the setup of it's mock interview from both the interviewer and the intervieweee's perspectives.

In the following section details of constitution and software workings of this thesis will be covered. First it will describe the general structure of the pipeline. Then it will describe how the inputs are processed from a bottom up approach; starting with raw sensor data and working up level by level to demonstrate how the engagement is calculated.

Chapter 2

Theoretical Background

2.1 Previous Work

[EXTENSIVE EXPLANATION ABOUT THE PREVIOUSLY READ PAPERS. CONCENTRATE MOST ON THE RICH PAPER]

2.2 Setup

2.2.1 Subject

The subject of our experiment is the interviewee in our mock interview. As shown in figure 2.1 the subject is seated aproximately 70 cm from a screen. A number of sensors are then set up to capture the needed inputs. Namely a Microsoft Kinect, an SMI Eyetracker and a microphone.

2.2.2 Agent

There are two main softwares used to simulate the virtual interview environment. First there is Charamel. Charamel is responsible for creating the interviewers (or agents) and their surrounding environment. The scene used for this thesis consists of two virtual characters, namely Curtis and Gloria. They stand behind a desk to mimic an office interview. On the left lies a white board that is used as an object in our environment. The setting is demonstrated in figure 2.2.

The second software used is Scenemaker. Scenemaker is responsible for sending the agents actions to perform. The program consists of a state

machine, each state containing a command to be executed by the agents accordingly. These commands include ordering the agents to utter a certain sentence, stop and wait for the subject to reply, perform a certain hand gesture, and so on and so forth.

Figure 2.1: Interview setup

Figure 2.2: Virtual environment

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 (Pointer) Meta data about the event.

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

State A boolean flag to indicate whether the event is starting or ending.

In the software's pipeline events are measuerd every time cycle. A cycle of 500 ms is used.

3.2 Sensors

3.2.1 Microsoft Kinect

[MORE DESCRIPTION]

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 prepetual displacement of the subject's head which indicates that s/he is nodding. This triggers an event called *HeadNod*. HeadNod is an event measured and outputed 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

[MORE DESCRIPTION]

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 of the subject's gaze point to the right alters the x coordinate and to the bottom affects the y coordinate.

The software defines two main rectangular 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 gaze 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. The event is prompted with pointer value 1 when the subject starts looking at the object, and triggered again with 0 when the subject directs his/her gaze away.

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 Voice Activity Duration) 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 2.2.2 Scenemaker is the software used to send commands to the virtual agents. Scenemaker is also responsible for sending the events that are triggered to represent 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 looks at the subject in front of the screen the event *AgentFacialGaze* 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

So to review our events, we have:

- HeadNod
- SubjectFacialGaze
- SubjectObjectGaze
- vad
- AgentFacialGaze
- AgentObjectGaze
- AgentSpeech

Those events will be used as inputs for our four main modules.

4.1 Mutual Facial Gaze

Mutual Facial Gaze is defined as the eye contact between the subject and the agent. It is necessary for the subject to direct his/her gaze at the agent's face when being addressed. The event *MutualFacialGaze* is triggered with a pointer value 1 (indicating that it started) when both SubjectFacialGaze and AgentFacialGaze are ongoing. When either of the two input events are triggered with the value 0 (event ends) the event MutualFacialGaze is also ends and therefore is triggered with pointer value 0.

4.2 Directed Gaze

Directed Gaze occurs when the agent points or looks at a certain object and then the subject follows. In our environment the white board acts as the object. The event *DirectedGaze* is triggered with pointer value 1 when both SubjectObjectGaze and AgentObjectGaze are ongoing. And triggered again with pointer value 0 when one of the two input events ends.

4.3 Adjacency Pair

Adjacency Pairs are usually defined as a speech utterance which is provoked by a previous speech utterance. For instance the answer to a question is an adjacency pair. This thesis did not go into the the semantics of natural language processing. The inputs where when the agent started and stopped speaking, and when the subject started and stopped speaking. Naturally we redefined the meaning of adjacency pairs to match our inputs.

This thesis defines adjacency pairs as a verbal statement uttered by the subject within a window of two seconds after the agent has finished speaking. It is assumed that any statement spoken by the subject shortly after the agent finishes is provoked by the agent's previous sentence. The *AdjacencyPair* start event is fired when a vad start event is detected within 2 seconds of an AgentSpeech end event. An AdjacencyPair end event is prompted as soon as said vad event ends.

4.4 Backchanneling

Backchanneling is the small responses given by the subject during the time where the agent is speaking. These responses indicate that the subject is following what the agent is saying. [CHECK HOW THE RICH PAPER DESCRIBES IT]

Here we introduce the concept of a *Backchanneling Pulse*. Since backchanneling has a very short duration that usually lasts only one event cycle which is not enough time to influence the bayesian network. So instead when a backchanneling event is provoked it is outputed on 5 consecutive event cycles with pointer value 1, 0.75, 0.5, 0.25 and 0 respectively, as shown in figure 4.1.

A BCPulse can be triggered by two different ways. It can be prompted when a HeadNod is detected during AgentSpeech. Or it can be set off when a vad event with a duration less than 2 seconds is detected during an AgentSpeech.

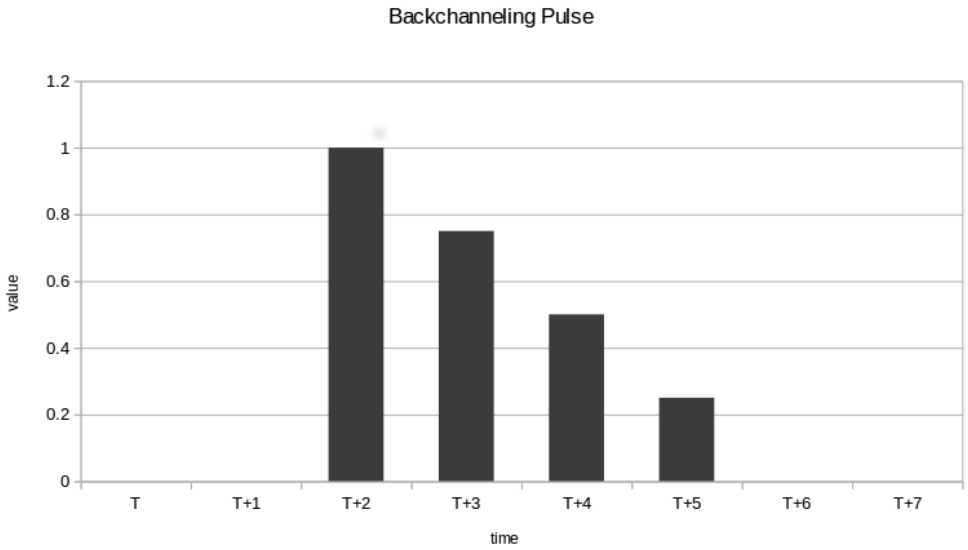


Figure 4.1: Backchanneling event triggered at T+2

Chapter 5

Bayesian Network

[ASK TOBY ABOUT HOW MUCH YOU CAN WRITE ABOUT THE BAYESIAN NETWORK. MINIMALLY JUST TALK ABOUT HOW THE ENGAGEMENT IS CALCULATED FEL AER]

Chapter 6

Summary

[EXPLAIN THE GENERAL STRUCTURE AGAIN IN MORE DETAIL THIS TIME. ADD IMAGE SHOWING YOUR SO CALLED BOTTOM UP APPROACH]

List of Figures

List of Tables

List of Algorithms