Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Dominance is relative: Automatic gaze annotation and conversational dominance in the MULTISIMO corpus

Lorcan McLaren
B.A. (Mod.) Computer Science and Language
Final Year Project, April 2020
Supervisor: Dr. Carl Vogel

School of Computer Science and Statistics
O Reilly Institute, Trinity College, Dublin 2
Ireland

Contents

List of Figures

List of Tables

Abstract

This paper proposes a method of automatic annotation of gaze behaviour in the MULTISIMO corpus that produce annotations for 1 minute of video footage in 2.5 minutes, compared to the 20 minutes needed by a human annotator. Comparison of the annotations produced by this system to manual annotation for the same session indicates 'good' inter-annotator agreement with a modified Kappa score of 0.79. Analysis of 210 minutes of footage across 7 sessions in MULTSIMO provides statistically significant results that suggest that dominant players spend more time looking at their fellow player than their less dominant counterparts, and avert their gaze for longer periods on average. Furthermore, the dominant individual receives longer gazes on average from the facilitator than their fellow player even though dominant and submissive players gaze at the facilitator at similar rates. The addition of gaze annotation, blinking annotation and head posture data covering 78% of the corpus should, hopefully, prove to be a valuable asset for future research.

Chapter 1

Introduction

1.1 Overview & Motivation

This paper will describe a heuristic-based approach to determining gaze allocation automatically in the MULTISIMO corpus, and establish whether a relationship exists between certain gaze behaviours and conversational dominance. Gaze annotation will also enable consideration of gaze behaviour in future research on addressee detection and turn-taking dynamics in this corpus.

Gaze is a proxy measure for estimating focus of attention, with humans making saccades (rapid eye movements) in order to 'bring areas of the visual world onto the fovea (the part of the retina with the highest acuity) for further scrutiny'(Hessels et al., 2019). Kendon (1967) states that gaze serves three primary functions: monitoring, expression and regulation.

Monitoring describes gaze used as a source of nonverbal information from interlocutors, whereas expressive gaze conveys information to interlocutors. For example, Argyle and Dean (1965) posit that mutual gaze duration is used to generate and manage intimacy.

Gaze also serves to regulate processes necessary for successful interaction, including indicating addressee and managing turn-taking. For example, in turn-taking, the speaker will link their utterance to what was previously said (thematic part). They will then gaze away while they continue to hold the floor, before gazing back towards the hearer as they [the speaker] provide a new piece of information (rhematic part). This gaze behaviour and content

together indicate the end of the turn (Heylen et al., 2002).

Social or conversational dominance is a factor that is related to the verbal and non-verbal behaviour of an individual, and their perception by others. Speakers who are conversationally dominant and either extroverted or disagreeable have been found to be more persuasive (Bee et al., 2010). Conversational dominance appears as a factor even in unstructured interaction where participants are unacquainted and of equal standing. Social background and gender can have an impact on dominance (Itakura, 2001).

The application of machine learning, and deep learning in particular, to computer vision problems has shown remarkable results in recent years. Deep learning performs particularly well when compared to rule-based approaches in unconstrained, real-world scenarios where the patterns in question may be too complex for a human to reason about. Examples where state-of-the-art results have been achieved using deep learning include object classification, image reconstruction and facial recognition, among others.

While this approach clearly has many merits, it requires a large amount of training data to build an effective model, larger than what is currently available in the MULTISIMO corpus. Furthermore, the simplicity and transparency of a rule-based system ensures that decisions about classification of gaze direction, for example, are explainable and allows it to be fine-tuned to participants of a session to a degree. This – along with a certain number of assumptions that we can make about our data due to the environmental constraints of the sessions in the corpus – enables the development of a system that performs with an adequately high level of accuracy for the use-case at hand. Moreover, the output of the system described below does not require much manipulation in order to produce annotations that can be imported to ELAN for use in future research.

The modelling of human-human interactions is imperative in working to develop artificial agents that can properly interpret the social cues necessary for successful and fluid exchanges. Embodied virtual agents will also need to manage their own behaviour to ensure the intended perception by humans is met. The investigation of factors influencing multiparty interaction, including conversational dominance, is important in furthering scientific understanding of social psychology. It is hoped that the results of this research as well as the gaze annotation produced for the MULTISIMO corpus will contribute to future research.

1.2 Related Work

A large body of research indicates that gaze behaviour is linked to conversational role i.e. whether an individual occupies the role of speaker or hearer. Kendon (1967) observes that hearers gaze at speakers more often than speakers gaze at hearers – with hearers gazing at speakers for longer durations with short breaks to gaze away – while speakers alternate been gazing away and gazing at the hearer at similar rates. Similarly, Argyle (1967) established that participants spent more time looking at their conversational partner when they were listening than when speaking. Malik et al. (2019) found that, in MULTISIMO, the person who the speaker was looking at is the addressee of the utterance in 77% of cases. However, this research was based on manual gaze annotation of only 2 sessions, so results should be taken as tentative.

Gaze behaviour may be linked to personality traits (Libby and Yaklevich, 1973). In a study where observers knew that they might be watched by the other later on, there is evidence that gaze behaviour also varies according to social hierarchy, with the mean duration an observer spent looking at higher ranked partner?s facial features in a photo being shorter than that spent looking at a lower ranked partner's features (Gobel et al., 2015). Foddy (1978) indicates that gaze behaviour also depends on whether the relationship between participants is competitive or cooperative.

Norris (2004) suggests that gaze behaviour is cultural, varying between social groups. Though no evidence is provided to support this, it is possible to imagine that individuals in some cultures might avoid eye contact with other individuals as a sign of deference or an act of modesty. Gobel et al. (2017) found that Western participants in their research were more likely to fixate on the eyes, while East Asian participants fixated on the bridge of the nose. It may be important to bear cultural factors such as this in mind in the context of the MULTISIMO corpus due to the heterogeneity of participants, with a wide range of nationalities represented.

Gobel et al. (2015) point out that the dual role of gaze in social monitoring and expression means that any results about gaze behaviour are context-sensitive and should only be generalised with caution. Norris also emphasises how gaze behaviour may be contextual, providing the example that gaze behaviour will be different in a static environment where two individuals face each other than in an environment where the individuals walk together in a forest. In a scenario such as that implemented in MULTISIMO, where

participants are aware they are under observation and the room is relatively distraction-free, the acts and words of their fellow participants are the only stimuli. We can assume that it will be 'sequentially structured' (Norris, 2004).

Bee et al. (2010) explore the impact of certain verbal and non-verbal cues on dominance perceptions, by experimenting with two models with different parameters of each of a number of behaviours. In terms of gaze, in the dominant model the IVA stares at a user while speaking, whereas it gazes for mean duration of 500ms and randomly switches between gazing at the user and averting gaze in the submissive model. The submissive model also has a bowed head, which has been found to reduce the perception of dominance. They found that gaze had an influence on the perception of extraversion but not directly on social dominance.

In the VACE corpus, gaze frequency – especially when duration was brief – was an indicator of passivity and has a negative correlation with dominance. Long gaze duration, however, was positively correlated with dominance (McNeill, 2005).

Fukuhara and Nakano's (2011) work involving the Wizard of Oz experiment found that more dominant speakers have higher turn-releasing success ratio. They also found that when speaking, an individual respected the dominant participant more as a hearer, providing more gaze attention to them than to a less dominant fellow participant. When acting as a hearer, an individual was also more likely to exchange mutual gaze with the more dominant person who was speaking, than with a less dominant person who was speaking.

Costello (2018) investigated gaze behaviour in MULTISIMO but did not perform significance testing due to availability of gaze annotation for only two sessions, meaning results were purely exploratory.

The remainder of this paper is organised as follows: Chapter 2 describes the corpus and development of the annotation system, as well as the evaluation of its performance and analysis of the data generated. Chapter 3 presents the results of this analysis, while Chapter 4 discusses these findings. Chapter 5 offers a conclusion and some perspectives on possible future work.

Chapter 2

Materials & Methods

This section will describe the methods involved in this research, including an overview of the corpus and the software libraries used, the processes involved in the video analysis, the subsequent creation of annotations, and a description of the evaluation of the performance of the system as well as the analysis of the data extracted from the image processing stage.

2.1 Resources

The research described in this paper makes use of the MULTISIMO corpus as well as an annotation tool known as ELAN and two Python libraries used in the video analysis component of the automatic generation of gaze annotation.

2.1.1 MULTISIMO Corpus & ELAN

MULTISIMO is a multimodal, multiparty corpus developed in Trinity College Dublin. It 'was developed to enable the understanding of human social behaviour in multiparty interaction, the structural representation of the interaction flow, and the modelling of the social communication mechanisms to be integrated into intelligent collaborative systems exhibiting natural behaviour' (Koutsombogera and Vogel, 2017).

The corpus is made up of sessions involving three participants, with two participants acting as players and the other acting as facilitator of the ses-

sion. The participants play a game intended to elicit cooperation among the players, similar to the popular American game show Family Feud. The facilitator asks a series of questions and players must come up with the 3 responses for each, ranked from most likely to least likely, that they believe will be the most common answers in a survey of 100 individuals when asked the same question. The facilitator provides feedback and guidance until the correct responses and order is achieved.

The corpus contains audio and video recordings from 18 different sessions involving 36 randomly allocated pairs of players, and three individuals who act as facilitators for each session. It includes individual video recordings showing the head and neck of each of the players, a 360° recording from a tabletop camera, and a recording that captures the whole scene including both the players and the facilitator (a cropped version of this that isolates the facilitator is also available). Stereo audio is captured by an omnidirectional microphone, in addition to audio coming from the head mics of each participant. Figure 2.1 shows the positions of facilitator, players and cameras for each session.

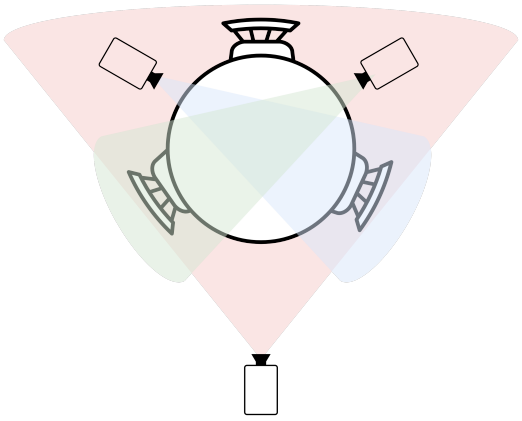

Figure 2.1: Plan of the room setup in MULTISIMO sessions

High quality video files with a resolution of 960x540 are available for the two players as well as the cropped recording showing the facilitator which has the same resolution but is of a lower quality. All cameras involved in the setup shoot at 30 frames per second.

A variety of different annotations are available for sessions within MULTISIMO including speech transcription, laughter annotation and facilitator

feedback annotation. These are the culmination of the efforts of a number of different researchers. Most of these annotations are in the form of EAF files, a filetype created and viewable using an annotation tool known as ELAN (Tacchetti, 2018) – developed by the Max Planck Institute for Psycholinguistics in the Netherlands.

Annotations in ELAN have 4 important components: the start time, the end time and the duration of the period of the annotation (all in milliseconds), as well as the actual annotation value itself. The goal of the video analysis component of this research is to produce annotations that may also be viewed and analysed in ELAN.

Existing Gaze Annotation

Manual gaze annotation currently exists for two sessions within the corpus as a result of previous work (Costello, 2018). The goal of the video analysis and annotation steps of this research is to replicate the format of these existing annotations as faithfully as possible, while also using them as a reference for validation of the automatically generated annotations – though existing annotations will not necessarily be used as a gold-standard of accuracy for reasons that will be discussed later in this paper.

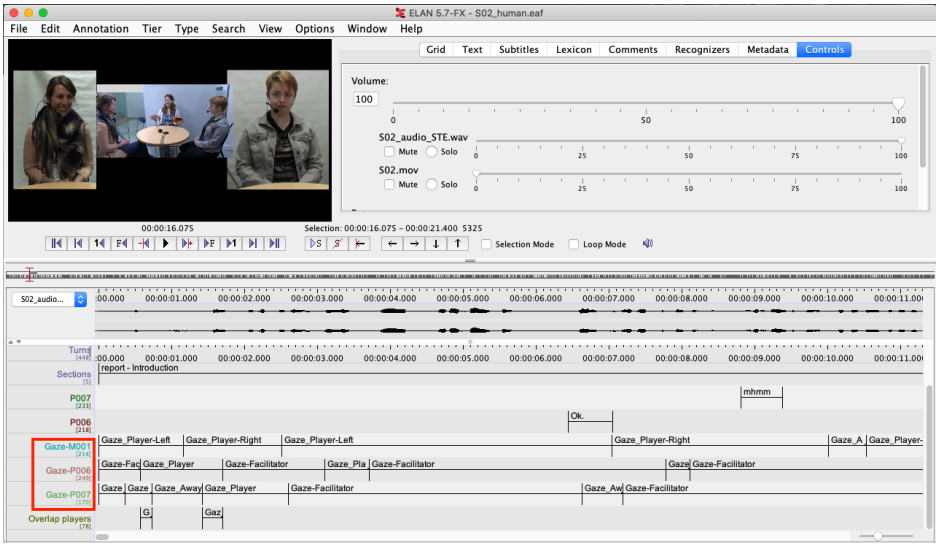

Figure 2.2: The ELAN interface for annotation

The existing gaze annotation has a controlled vocabulary, meaning possible annotation values are constrained to a predefined list. The following values are possible for each of the players:

- Gaze_Player
- Gaze-Facilitator
- Gaze_Away

An annotation tier also exists for the facilitator, which has its own vocabulary that includes the 'Gaze_Away' annotation value as well as the following two relating to the subject of the facilitator's gaze:

- Gaze_Player-Left
- Gaze_Player-Right

Note that throughout this paper, left and right will correspond to the perspective of a viewer who faces the facilitator/player in question. This is standard practice in this corpus based on the precedent set by existing gaze annotation for facilitators.

Human Dominance Assessment

A human dominance assessment was provided for each player in order to generate ground truth for Costello's (2018) work on establishing linguistic indicators of conversational dominance. They will also be the source of ground truth for this work, against which gaze behaviour can be compared. 5 annotators viewed each of the 18 sessions twice and assigned a score to each player on a scale from 1, being least dominant, to 5, being most dominant. The annotators were not informed as to what verbal or non-verbal cues might be indicators of dominance, but they were provided with a definition of conversational dominance as 'a person's tendency to control the behavior of others when interacting with them' (Koutsombogera and Vogel, 2017)

2.1.2 OpenCV

OpenCV is an open source image processing library built-in C++ that is intended to provide access to powerful computer vision techniques that work in close to real time.

Images in OpenCV are represented as two-dimensional matrices, where each element constitutes a pixel. In RGB colour space, this element will be a tuple where each value in the tuple represents a shade of red, green and blue respectively. Many computer vision techniques only function in grayscale colour space, where each pixel may be represented by a single integer value, thereby reducing dimensionality that would have been prohibitively expensive in terms of memory and processing time for earlier computers.

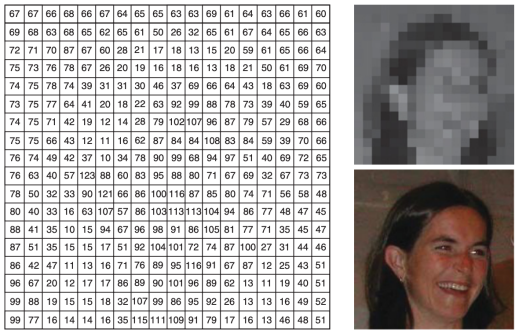

Figure 2.3: Matrix representation of an image (Dawson-Howe, 2014)

Video in OpenCV is analysed as a sequence of images, and therefore all the operations described in the following sections must be applied to every single frame individually.

This can lead to results that are sometimes imperfect as each image is analysed without taking the preceding or succeeding frames in account. However, as will be discussed in Section 2.3.1, with some clever filtering, we can reduce the overall error rate and improve the quality of the resulting annotations.

2.1.3 Dlib

Dlib is a C++ software library that provides implementations of machine learning tools and pretrained models for a wide variety of applications including image processing. In our use case, the frontal face detector is used

to detect the faces in an image while the shape predictor is used to project a number of facial landmarks onto the located face that will be used for all of the steps described in Section 2.2.

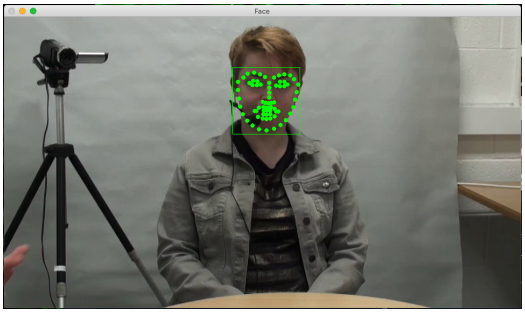

Figure 2.4: Dlib's facial landmarks in action

These 68 facial landmarks allow for the tracking and isolation of certain points and regions of the face.

2.2 Video Analysis

Each of the following 3 steps of video analysis is contingent on the accurate identification and isolation of faces. Initial attempts to implement face and eye recognition were made using the pre-trained Haar cascade classifiers available within OpenCV. However, their performance on individual frames varied, with the eye classifier in particular having trouble, often misidentifying the mouth as an eye, for example. A number of resolutions for these issues were tested, including stipulating that eyes could only appear in the upper half of a face and that there must be exactly 2 eyes per face. However, after a period of trial and error, it was found overall performance was improved when Dlib's frontal face detector was used instead. As previously mentioned, many computer vision techniques only function in grayscale colour space, so it was necessary to first convert the image to grayscale before using the face detector. All further image analysis operations described in the following subsections were also performed on frames in grayscale format.

Following facial identification, the 68 facial landmarks were projected onto the located face using Dlib's shape detector. The most significant landmarks for the subsequent stages of analysis and their relevant index were:

- Chin tip: 8
- Nose tip: 33
- Left eye region: 36-41
- Right eye region: 42-47
- Mouth corners: 48 & 54

It is worth mentioning at this time that none of the techniques described below are failsafe, with visual information being among the most complicated, unconstrained and information-saturated forms of data to deal with, alongside language. Performance may be impacted by lighting conditions and angles of seats among other factors. These issues will be given further treatment in the discussion of results.

2.2.1 Blink Detection

Each eye region was isolated by cropping each frame using the location of the landmarks described above. The Euclidean distance between the two landmarks indicating the inner and outer corner of the eye was calculated.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.1}$$

No landmarks indicate the centre of the upper and lower eyelid, but this was approximated by calculating the midpoint between the two landmarks provided for each lid. Once again, the Euclidean distance between the two midpoints was found. Dividing the first distance by the second produces a ratio that will here be referred to as the 'blinking ratio'.

The reason a ratio is necessary rather than simply calculating the number of pixels between the upper and lower eyelid is that this ensures blinking detection is scale-independent and will function correctly regardless of the resolution of the source video or the proximity of the participant to the camera. This blinking ratio was calculated independently for each eye and then averaged. The average value was used to determine whether a participant was blinking, with a 'blinking' classification being assigned to a frame if the ratio value exceeded a specified threshold.

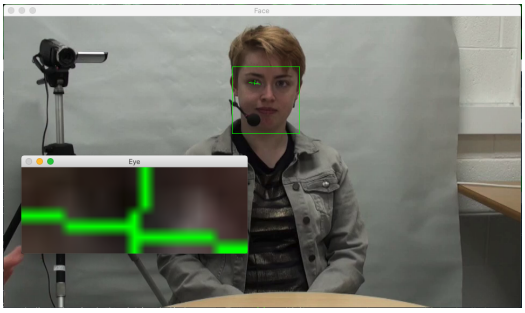

Figure 2.5: Calculating the blinking ratio

Blink detection was a necessary step both for preventing an error where the eyes could not be found if closed in a given frame – which caused the gaze direction analysis step to fail – as well as for the reason that this data may prove to be useful in the context of this and/or future research.

2.2.2 Gaze Direction Analysis

Once again, each eye region was isolated using the facial landmarks in the same manner as the previous step. Initially, an attempt was made to mask the region surrounding the eye itself in order to exclude parts of the lids that had been included when the eye was isolated (as all images in OpenCV must be rectangular meaning some excess must be allowed for). However, based on the observation of test runs via webcam and clips from MULTISIMO sessions, early performance needed to be improved, and removing this mask was one change that led to a marked improvement.

The image was then thresholded to separate the two regions of interest in the eye – the iris and the sclera (or 'white' of the eye) – by creating a binary image. Binary thresholding is an image operation where each pixel in an image is mapped to the maximum or minimum possible value (white and black respectively, in grayscale colour space) based on its initial value in relation to a specific threshold. For example, if a threshold of 127 were chosen, all pixels with a value in the upper half of the grayscale spectrum (i.e. above the threshold) would be mapped to white while all values in the lower half (i.e. below the threshold) would be mapped to black. Otsu's method dynamically determines an optimum threshold based on the image histogram. It performs well for 'bimodal' images, which is to say images where a high

contrast exists between the regions we wish to separate. In OpenCV, the threshold value chosen is the one that maximises the between class variance σ_B^2 for the 'foreground' and 'background' regions (Dawson-Howe, 2014) i.e. the iris and sclera in our case.

$$\sigma_B^2(T) = w_f(T)(\mu_f(T) - \mu)^2 + w_b(T)(\mu_b(T) - \mu)^2 \tag{2.2}$$

The above equation is used to calculate this between class variance for the f (foreground) and b (background) classes, where T is the threshold value and μ and σ^2 are the mean value and variance of the image respectively.

A number of operations were used to refine the newly formed binary regions. Erosion reduces the number of object pixels by removing pixels from the perimeter of the foreground, ensuring a precise edge around the region of interest. Dilation expands the number of object pixels, filling any small gaps that may appear within the region of interest. Together, these two operations improved the quality of the binary image by ensuring the foreground and background were cleanly separated into uninterrupted regions with precise boundaries.

The eye region was then split into left and right halves in a similar manner to that used for blinking detection. The midpoints between the two landmarks on each lid were calculated separately and the line between these two midpoints was used to divide the eye region.

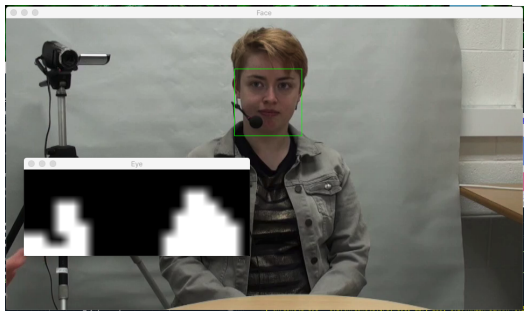

Figure 2.6: Thresholding the eye region

It was then necessary to count non-zero elements in each half. Since white has a value of 255 and black has a value of 0 in grayscale colour space, this is equivalent to counting the white pixels. Finally, the ratio of the returned

value for the left side of the eye with respect to that of the right side was found.

Gaze direction was classified into three ordinal directions – left, centre and right – on the basis of this value through use of an upper and lower threshold. If the ratio value was less that the lower threshold for a given frame, the participant was classified as gazing left. If the value lay between the upper and lower thresholds, the participant was classified as gazing ahead, or to the 'centre'. And if the value exceeded the upper threshold, the participant was classified as gazing right.

The procedure for converting these ordinal conversions to annotations following the precedent set by existing manual gaze annotation is described in Section 2.3.

2.2.3 Head Posture Analysis

Head posture analysis enables a rotation vector to be calculated that indicates the direction a participant is facing. The solvePNP function of OpenCV calculates the relationship between points on a 2D image representation and their positions in a model of 3D space. In this case, the two sets of points used were the facial landmarks located in the image using Dlib, and the positions of these same landmarks on a hypothetical 3D model of a human head, where the origin (0,0,0) is found at the tip of the nose.

The landmarks used in this case were the chin tip, the nose tip, the two corners of the mouth and the outer corner of each eye. The correspondence between these sets of points was then calculated using solvePNP, which produces a rotation vector and translation vector.

The data produced in this stage of analysis was not used in the creation of gaze annotations, nor was it used in statistical analysis to establish a relationship between gaze behaviour and conversational dominance as the scale of the work required to complete other portions of this research was already extensive. However, it will hopefully serve as a useful addition to the MULTISIMO corpus for future work.

The results of head posture analysis are perhaps more useful when the resulting rotation vectors are converted to Euler angles. This requires the intermediate step of first converting the vector to a rotation matrix using the Rodrigues' formula.

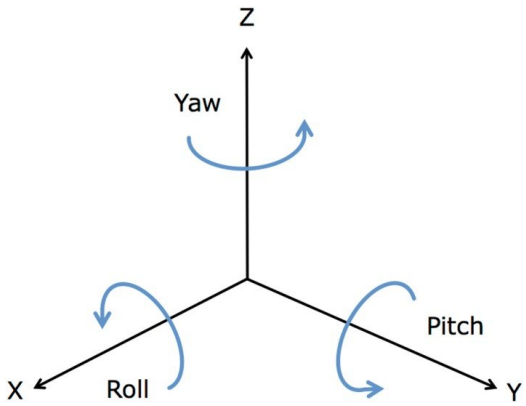

Figure 2.7: The Euler angles

Yaw indicates the left-right turn of the head. Provided with some labelled training data, a model could be developed that classifies head posture into left, centre and right categories in the same manner as gaze direction analysis, which may be used to augment the accuracy of this previous stage or, alternatively, to provide an additional dimension of data for analysis.

2.3 Annotation

The previous three steps produce a data frame containing a row for every single frame of each video with a column each for the relevant index, ordinal direction, rotation vector and timestamp (to give an idea of scale, there are 18,000 frames in a 10 minute video and 3 videos to be analysed per session). In order to produce annotations that are useful from this, a filtering and merging process is necessary. Remember that each annotation in ELAN must have a start time, an end time, a duration and an annotation value.

Originally, an estimated timestamp was being assigned to each frame by multiplying its index (i.e. the number of frames that have occured before it) by a constant value derived from our knowledge of the framerate of the cameras. However, initial annotations produced in this manner had a length mismatch with the video files, with each being shorter by varying degrees than they ought to be. Eventually it was discovered that this length mismatch resulted from the inability of OpenCV to read certain frames of video, potentially due to corruption, that led to the index for a frame being lower

than it should be. This caused a delay between the annotations and corresponding video sequence of video that worsened over the course of the video, as corrupted frames seem to be evenly distributed throughout. A resolution was eventually found in that OpenCV is able to obtain a timestamp directly from the source video. This was added to the dataframe in lieu of the manually calculated timestamps.

As previously mentioned, the intention of this step was to produce annotations that reflect the format of human gaze annotation as faithfully as possible, with the additional annotation value of 'blinking' for one of the types of annotation produced.

The first step of the process was to convert ordinal direction values of 'left', 'centre' and 'right' to the desired annotation value. This annotation value is dependent on the position of the participant in question to the other two participants, and therefore the Python script generating annotations from the video analysis data had to be modified for each video to ensure each ordinal direction was mapped to the correct annotation value. For example, in the case of a player found to the left of the facilitator, a gaze direction of 'left' should be mapped to 'Gaze-Facilitator' while 'right' should be mapped to 'Gaze_Player'. For a player found to the right of the facilitator, these annotation values should be switched. And for the facilitator, annotation values of 'Gaze_Player-Left' and 'Gaze_Player-Right' are used for the left and right ordinal directions respectively.

Subsequently, sequences of frames where the annotation value was the same were merged into a single annotation, with the start time being the timestamp for the first frame in this sequence, the end time being the timestamp for the last, and the duration being calculated by subtracting the former value from the latter.

2.3.1 Types & Filtering

As previously mentioned, vision remains one of the most challenging processes to attempt to replicate computationally, and no system comes anywhere close to human performance on tasks that come easily and naturally to us.

One of the issues faced with video analysis is a lack of continuous, contextual reasoning across a series of frames. Our eyes may be fairly easily

deceived momentarily, but our brains are generally quick to make inferences about reality based on 'commonsense' knowledge. A computer, by contrast, has no such domain knowledge – including no sense of what behaviours are normal – and therefore lacks an understanding that it is unlikely that a person went from gazing left to gazing right and back again in the span of 33 ms. Rather than trying to implement an algorithm that accounts for the results of preceding and succeeding frames at the image processing stage – which would be prohibitively difficult and technical for a project of this scope – it was found that with some simple heuristics bring the performance of the system to a level sufficient for our needs.

Two versions of annotation were created for each of the videos analysed. The first fine-grained annotation is a closer representative of the actual output of video analysis. All entries with a duration of 100ms or less (approximately 3 frames of footage) were removed on the basis that there was a reasonably high probability that these could be misclassifications and, even if it were possible to filter out these misclassifications, such fleeting annotations were unnecessary for the purposes of this paper. Entries where the value was 'blinking' were maintained regardless of brevity, as blinking is evidently a very rapid process.

Broad-grained annotations were produced at a later stage that are a better approximation of the type of annotation a human annotator might provide. Entries where value was 'blinking' were removed, and neighbouring entries extended in order to fill the resulting gaps. This step was necessary both for the assessment of inter-annotator agreement – as the lack of annotation for blinking in the human annotations would make comparison difficult – and for the purpose of the analysis described in subsequent sections, as blinking behaviour disrupts the continuity of gaze annotation in a way that is incompatible with the metrics used. It would, however, be very easy to transfer this blinking annotation to another tier with ELAN if required in future.

2.3.2 Importing into ELAN

The CSV files generated by the annotation phase can be viewed in ELAN using the built-in functionality for importing this file type. Then it is relatively simple to add the relevant video source as a linked file and save the project as an EAF file, enabling all research and analysis to be performed in the same manner as with a human annotation.

2.3.3 Inter-Annotator Agreement

The reliability of annotation was assessed by comparison with a human annotation for the same session. Note that granularity difference resulting from frame-by-frame analysis – a level of detail that would not be feasible in manual annotation – means that it is not necessarily a gold-standard here but rather a benchmark for comparison and validation that the resulting analysis of the data produced is consequential.

ELAN offers a multiple file processing facility, including assessment of inter-annotator agreement between two EAF files. In this case, a modified Kappa statistic with a minimum of 60% overlap required was used. Raw agreement is insufficient for assessing reliability. If, for example, there are two possible annotation values and one occurs more frequently than the other in ground truth, an annotator could simply provide this more common value for all annotations and achieve a relatively high raw agreement score. By contrast, Cohen's Kappa is a chance-corrected agreement index that 'normalizes the observed agreement by the amount that could be expected by chance alone' (Holle and Rein, 2015).

Even for humans, separating something continuous like the angle of gaze into discrete categories – such as left, centre and right – is difficult, with the exact point where one category meets another varying between annotators, so perfect agreement was not expected here.

2.4 Data Analysis

The annotations for 7 of the total 18 sessions were chosen for statistical analysis. 3 separate video files had to be annotated for each session (one per participant) and sessions lasted 10 minutes on average, meaning the following statistical analysis is based on approximately 210 minutes of video footage. The execution of the video analysis was observed for all 54 video files, and these 7 sessions were chosen on the basis of this author's assessment that classification of gaze direction was of at least as high a quality as those that would be produced by a human annotator. Reasons the other sessions were excluded include: adverse lighting conditions, angles of chairs, and the positions of participants relative to cameras.

2.4.1 Human Dominance Assessment

A human dominance assessment for each player in MULTISIMO is included in the corpus data. 5 annotators watched each session twice and provided a dominance score for each participant on 5 point scale, with 1 being the lowest and 5 being the highest (Vogel et al., 2020).

An intraclass correlation coefficient [ICC] was used to establish the level of agreement among annotators. The ICC was determined to be 0.776, which indicates 'good' agreement (Costello, 2018).

Averages across these 5 assessments are used to calculate an overall dominance score for each participant for the purposes of this paper.

2.4.2 Metrics

There are four different types of gaze acts of interest for each session, characterised by the specific combination of gazer issuer and recipient in each case.

- Facilitator→player gaze
- Player→player gaze
- Player→facilitator
- Player→away

In establishing the human dominance assessment, the 5 annotators were informed that 'dominant individuals both move and talk more, and do so more often than non-dominant people' (Costello, 2018). This contributed to the decision to measure both frequency and average duration in relation to the four gaze acts listed, with the expectation that there ought to be some statistically significant relationship between conversational dominance and some of these metrics that supports the use of gaze behaviour as an indicator of dominance.

The results of a Shapiro-Wilk test for normality led to the choice of non-parametric tests in the following sections.

2.4.3 Absolute Dominance Score Correlations

A Spearman correlation test was used to determine whether a relationship existed between absolute dominance score (i.e. the average score obtained across the 5 annotators) and mean duration of the three of the gaze acts listed in the previous subsection: player→player, player→facilitator, and facilitator→player gaze.

2.4.4 Relative Dominance Ranking

In addition to the relationship with absolute dominance score, it was considered necessary to explore if there was a relationship between the four gaze acts listed in the previous subsection and the relative dominance of a participant, which is to say their 'rank' when players are split into two groups based on their dominance score. For each session, the player with the higher dominance score was assigned to the R1 group while the player with the lower dominance score was assigned to the R2 group.

2 of the 7 sessions chosen for statistical analysis had players whose dominance scores were identical. In these cases, players were split into left and right groups based on their position in the room in relation to the facilitator. The hope was that this could act as a control to compare against the ranked groups.

For the duration and frequency of each of the four gaze acts, an unpaired Wilcoxon test was used to evaluate the significance of the relationship between ranked group membership and the gaze act in question e.g. if there is a statistically significant difference between the mean duration of player→facilitator gaze for R1 versus R2 participants.

For all tests of statistical significance, the null hypothesis is rejected only if $p <= \alpha$, where $\alpha = 0.05$ in the context of this research.

Chapter 3

Results

3.1 Annotation Quality & Corpus Coverage

11 sessions were excluded from analysis on the basis of inadequate performance on 12 video files. Unfortunately, a session had to be entirely disregarded in the context of this research if even 1 of the 3 videos posed an issue for automatic annotation. However, automatic annotation was performed successfully for 42 out of 54 total video files, leading to an overall coverage of approximately 78% of the corpus.

As mentioned in Section 2.4, performance of the system in these cases was affected by adverse conditions. Misclassifications may broadly be organised into two categories:

1. The first category encompasses misclassifications resulting from true failures of the system to adequately locate the position of the iris. Causes included glare on the glasses of a participant – which prevented proper thresholding of the eye region –, occlusion of the eye region by the frames of glasses, and deep shadows cast over the eye region resulting from overhead lighting and the bowed head posture of participants.

2. The second source of misclassification results from the relative angles and positions of participants and cameras. In a number of sessions a camera and/or the chair of a player had moved, meaning that, although the system was adequately able to locate the iris and therefore determine the direction of gaze, problems arose in the conversion of

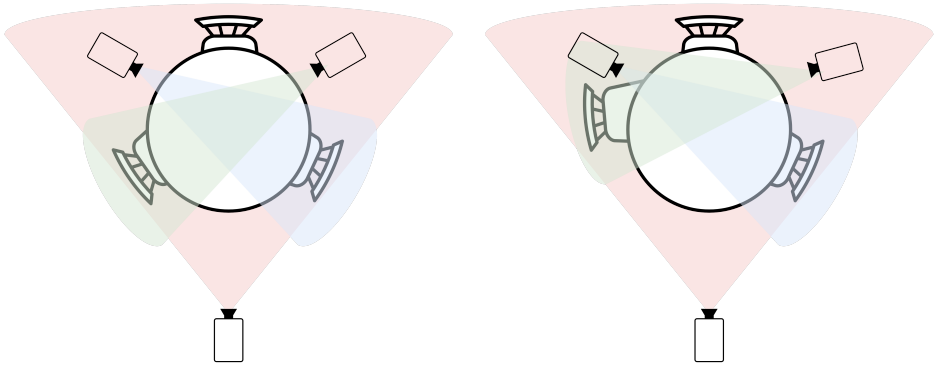

Figure 3.1: A comparison of room setups – incorrect setup on right

ordinal directions to annotation values. As usual, frames where the iris was found in a central location were annotated as 'Gaze_Away', while frames where the iris occupied the left side of the pupil were annotated as 'Gaze_Player' (where the player being analysed was found to the facilitator's right). However, due to the closer proximity of the camera (filming the player being analysed) and the other player, the movement of the iris between a central position – which ought to be annotated as 'Gaze_Away' – and a central position – which ought to be annotated as 'Gaze_Player' – was not significant enough to be detected. This resulted in many misleading annotations for the affected participant.

3.1.1 Inter-Annotator Agreement

Inter-annotator reliability was used to measure the agreement between automatically generated annotations for Session 2 and the corresponding manual annotations. Manual annotations were also available for Session 18 which could have been used for comparison, however, this session had to be excluded from analysis due to lighting conditions impairing performance for Participant 39 in particular. Shadows on the eye region created by overhead lighting meant that establishing a threshold value that could adequately separate the iris from the sclera in a binary image was not possible, preventing accurate gaze annotation.

As mentioned in Section 2.3.3, Cohen's Kappa is a metric that normalises the observed agreement by the probability of chance agreement, with 0 indi-

cating that the two annotations in question are in total disagreement, while 1 indicates total agreement. A variant of Cohen's Kappa requiring a minimum of 60% overlap for a match to occur was used here.

Participant	**Kappa**	**Raw Agreement**
P006	0.8416	0.9028
P007	0.6629	0.7841
Facilitator	0.875	0.9338
Average	*0.7932*	*0.8736*

Table 3.1: Modified Kappa results for Session 2

The results of inter-annotator agreement assessment excluding unlinked values are presented in Table 3.1. The average Cohen's Kappa for the three participants in Session 2 is 0.79, with a raw agreement value of 87%. Some researchers suggest that Kappa value of from 0.6 to 0.79 indicates substantial agreement while 0.8 to 1 indicates almost perfect agreement. Our value is found precisely on the cusp of the two. Others, however, are more demanding, however, with a Kappa value of 0.8 or higher providing good reliability, while a value of 0.67 to 0.79 allows tentative conclusions to be drawn (Artstein and Poesio, 2008).

While our Kappa value indicates an annotation on the lower end of 'good' reliability by this definition, it is sufficiently accurate for conclusions about the relationship between gaze behaviour and dominance to be drawn that ought to be significant enough to make inferences about the general population.

3.2 Human Dominance Assessment

The preexisting dominance scores by the 5 annotators were averaged for each of the 7 sessions. These average scores can be seen in Table 3.2.

Table 3.3 shows the membership of each group after this step after players have been divided into R1 and R2 groups based on the relative average dominance score of their paired player.

Session	Player 1 (Left)	Player 2 (Right)
S02	2	3.4
S05	3.8	2
S07	3.4	3.4
S09	3.6	3.6
S20	4.2	4
S22	3.4	3.6
S23	3.6	3.4

Table 3.2: Average dominance score for sessions analysed

Session	R1	R2
S02	P007	P006
S05	P013	P012
S07	Equal	Equal
S09	Equal	Equal
S20	P042	P043
S22	P047	P046
S23	P048	P049

Table 3.3: Group membership based on average dominance score

3.3 Absolute Dominance Score Correlation

Table 3.4 shows the figures used in the Spearman correlation analysis for each of the 3 gaze acts under investigation: facilitator→player, player→facilitator and player→player gaze (for each player in a pairing)

Player	**Score**	**F→P**	**P→P (Rec.)**	**P→F**	**P→P (Iss.)**
P006	2	2164.048	2742.193	2736.408	2505.657
P007	3.4	2638.179	2505.657	4058.014	2742.193
P012	2	1371.914	4500.156	3307.884	1332.968
P013	3.8	2369.209	1332.968	2785.056	4500.156
P016	3.4	1121.916	2513.79	3394.737	713.2823
P017	3.4	1698.238	713.2823	4337.025	2513.79
P020	3.6	2078.725	1367.642	2297.85	1479.296
P021	3.6	1714.19	1479.296	3079.966	1367.642
P042	4.2	2456.205	1526.615	1337.967	1544.758
P043	4	1925.946	1544.758	1293.273	1526.615
P046	3.4	2489.099	946.1905	1093.375	1141
P047	3.6	2064.806	1141	1147.928	946.1905
P048	3.6	2620.46	1837.081	445.2429	1925.975
P049	3.4	2555.273	1925.975	2556.394	1837.081

Table 3.4: Statistics used for correlation: mean duration (ms) of gaze acts

Figure 3.2 shows a plot of the correlation between the mean duration of facilitator→player gaze and dominance score. While the regression line exhibits a general upward trend, with mean duration of this gaze act increasing as dominance score increases, no statistically significant relationship was found ($p = 0.6359$).

Figure 3.3 shows a plot of the correlation between the mean duration of player→player gaze and the dominance score of the gaze-receiving player. The regression line exhibits a downward trend, with the mean duration of gazes received from another player reducing as dominance score increases. However, though more noteworthy than the previous relationship investigated, this is also not statistically significant ($p = 0.1077$).

Figure 3.4 shows a plot of the correlation between the mean duration of player→facilitator gaze and dominance score. The regression line again exhibits a slight downward trend, with the mean duration of gazes received

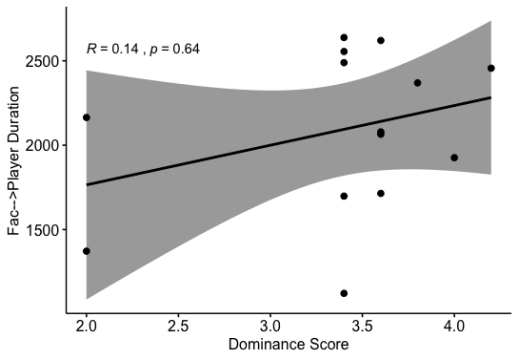

Figure 3.2: Scatter plot of gaze received from facilitator vs. dominance score

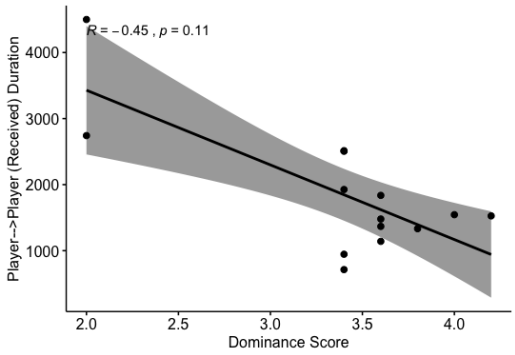

Figure 3.3: Scatter plot showing gaze received from player vs. dominance score

from another player decreasing as dominance score increases. The relationship again is not statistically significant ($p = 0.1057$).

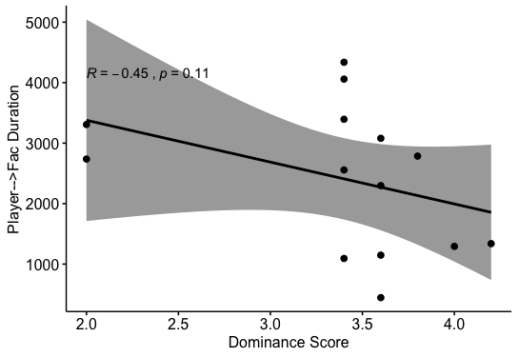

Figure 3.4: Scatter plot showing gaze given to facilitator vs. dominance score

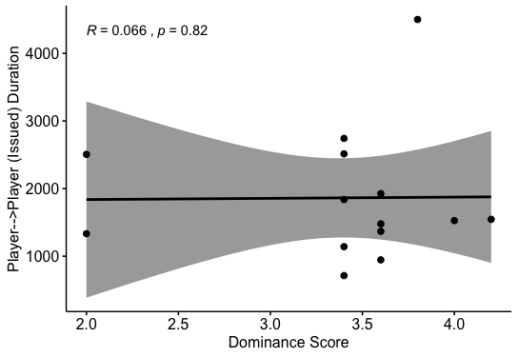

Figure 3.5: Scatter plot showing gaze given to player vs. dominance score

Figure 3.5 shows a plot of the correlation between the mean duration of player→player gaze and the dominance score of the gaze-issuing player. The regression line is close to flat in this case, meaning there is almost no reason to believe that the null hypothesis should be rejected ($p = 0.8226$).

3.4 Ranked Relative Dominance Analysis

In these tests, for each session, the more dominant participant according to their mean dominance score is assigned to the R1 group while the less

dominant one is assigned to the R2 group. The results presented in Section 3.4.1 are for the 5 sessions where there is a difference in the mean dominance score for participants. There are 2 sessions where participants have equal dominance scores. These will be discussed in Section 3.4.2.

3.4.1 R1 vs R2 Players

Player→Facilitator Gaze

The mean duration of player→facilitator gaze is 1832.082 ms for R1 members, whereas it is 2080.713 ms for R2 members. The difference in the mean duration for this gaze act between groups is not significant ($p = 0.473$).

The mean frequency of player→facilitator gaze is 95 per session for R1 members, whereas it is 98.8 per session for R2 members. The difference in the mean frequency for this gaze act between groups is not significant ($p = 0.5476$).

Player→Player Gaze

The mean duration of player→player gaze (where the player in question is the gaze-issuer in each case) is 2309.762 ms for R1 members, whereas it is 1647.031 ms for R2 members. The difference in the mean duration for this gaze act between groups is significant ($p = 0.0000001339$).

The mean frequency of player→player gaze (where the player in question is the gaze-issuer in each case) is 91.6 per session for R1 members, whereas it is 102.8 per session for R2 members. The difference in the mean frequency for this gaze act between groups is not significant ($p = 0.2492$).

Facilitator→Player Gaze

The mean duration of facilitator→player gaze is 2433.452 ms for R1 members, whereas it is 1982.294 ms for R2 members. The difference in the mean duration for this gaze act between groups is significant ($p = 0.03043$).

The mean frequency of facilitator→player gaze is 94.6 per session for R1 members, whereas it is 97.2 per session for R2 members. The difference in the mean frequency for this gaze act between groups is not significant ($p = 1$).

Player→Away Gaze

The mean duration of player→away gaze is 709.8353 ms for R1 members, whereas it is 590.5616 ms for R2 members. The difference in the mean duration for this gaze act between groups is significant ($p = 0.001415$).

The mean frequency of player→away gaze is 102 per session for R1 members, whereas it is 102.2 per session for R2 members. The difference in the mean frequency for this gaze act between groups is not significant ($p = 1$).

3.4.2 Equal Dominance Players

In addition, in 2 sessions the participants have equal dominance scores. In these cases, players were split into L- and R-Player groups based on their position relative to the facilitator.

The mean duration of player→facilitator gaze is 3456.806 ms for L-Players, whereas it is 3277.196 ms for R-Players. The difference in the mean duration for this gaze act between groups is not significant ($p = 0.918$).

The mean duration of player→player gaze (where the player in question is the gaze-issuer in each case) is 2030.572 ms for L-Players, whereas it is 942.822 ms for R-Players. The difference in the mean duration for this gaze act between groups is significant ($p = 0.0000006457$). The mean duration of facilitator→player gaze is 1819.782 ms for L-Players, whereas it is 1311.805 ms for R-Players. The difference in the mean duration for this gaze act between groups is significant ($p = 0.008231$).

The mean duration of player→away gaze is 1059.142 ms for L-Players, whereas it is 1007.633 ms for R-Players. The difference in the mean duration for this gaze act between groups is not significant ($p = 0.5706$).

While it may be disconcerting that a significant difference is found between these L- and R-Player groups for the mean duration of 3 gaze acts, this could be heavily skewed by the behaviour of a single participant and does not provide good basis for inference as only 2 sessions of equal dominance were analysed.

Chapter 4

Discussion

78% coverage of the corpus will hopefully prove to be useful for future research even if it is not for the purposes of this research. This could be augmented by providing manual annotations for the 12 problematic video files in future to ensure 100% coverage is achieved.

Manual annotation of gaze is an extraordinarily labour-intensive process, with Costello (2018) observing that it took approximately 20 minutes to annotate 1 minute of footage with gaze information for the MULTISIMO corpus. Based on this, it would have taken approximately 70 hours to annotate the 210 minutes of video footage whose annotation is used for statistical analysis in this research. By contrast, it takes approximately 2.5 minutes to provide annotation for 1 minute of footage using this automatic system, representing an enormous reduction in time over manual annotation. This enables large quantities of data to be produced quickly and reliably, with only light supervision required at the beginning of analysis of each file to ensure environmental factors are not preventing accurate classification. In this regard, it is highly useful as a tool to speed up annotation even in existing corpora.

As mentioned, quality of annotation and therefore utility of an automatic rule-based system such as this is contingent on certain assumptions about the environment, and its robustness is challenged in adverse conditions such as poor lighting, inconsistent positioning of chairs, and camera angles in certain sessions, as well as by glare produced by the reflection of light on the glasses of participants in some cases. It is possible that 100% coverage could be achieved with automatic gaze annotation in a corpus developed

with knowledge of this system's constraints in mind. While the resulting data is evidently not as precise as that produced by gaze-tracking eyewear, for example, it does allow rapid and low cost annotation (both in terms of labour and computational resources) without the need for specialist equipment.

In relation to the Kappa measure of 0.79, while this is only based on one session, it provides validation that the system worked correctly in this case and suggests that the performance in other 6 sessions would be similarly high, given that these were hand-picked based on observed accuracy. There was a fairly significant number of unlinked annotations between the manual and automatic versions as there were 633 annotations provided across the three participants in the manually annotated version versus 779 annotations by the automatic annotator (representing a 23% increase). This granularity difference will almost certainly have suppressed the Kappa statistic to a degree, meaning accuracy could indeed be even higher than measured.

On the whole, the sample used in this study is likely not large enough to investigate conclusively whether relationships exist by correlation, given that only a few participants appear in each numerical band. The annotation of dominance score was by nature comparative – as annotators will weigh the dominance score assigned to a player in relation to that assigned for the other player in the same session – so to compare these scores across sessions is also not necessarily a fair representation. Furthermore, the definition of dominance that was provided to annotators was broad and ambiguous (intentionally), leaving a lot of room for subjectivity. Neither Costello (2018) nor Vogel et al. (2020) specify whether these annotators worked linearly – with an annotator scoring each player in a session before moving on – or whether they were allowed to return to a video at a later stage. This means that the score provided could potentially not only be subjective to the annotator but also the result of developing criteria that evolved as the annotator completed their task.

A degree of chance was also involved in creating the ranked dominance groups, as some members who were in the R1 group would have been in the R2 group in a different pairing and vice versa. Calculating the variance for duration and frequency of each of the gaze acts within each group was considered, however, due to the differences in lengths of sessions – with some lasting under 7 minutes while others exceed 12 minutes in length – this would not have been useful. It might be interesting to explore if and how the results differed if the frequency was normalised by the length of the session.

Based on the results of the ranked relative dominance analysis, it is possible to adopt a number of alternative hypotheses and draw some statistically significant conclusions.

There exists a relationship between ranked dominance group membership and player→player gaze, with R1 players gazing at R2 players for a longer duration on average than the inverse. R2 players gaze at their fellow player more frequently than R1 players in the 5 sessions analysed, however, the p-value for this is not sufficiently low for us to rule this out as a chance result and therefore the null hypothesis may not be rejected i.e. there is no significant difference in the frequency at which R1 and R2 members gaze at their fellow players. Consequently, according to this analysis, dominant players spend more time looking at their fellow player than their less dominant counterparts.

A relationship exists between ranked dominance group membership and facilitator→player gaze, with the facilitator gazing at R1 players for a longer duration on average than the inverse. The mean frequency of this gaze act is very similar between R1 and R2 groups and a high p-value indicates that the null hypothesis should be maintained i.e. there is no significant difference in the frequency at which the facilitator gazes at R1 versus R2 players. Therefore, it may be affirmed that the facilitator spends more time gazing at the dominant player than at the other, less dominant player. It is worth considering whether the facilitator gives more attention to the R1 players as a result of their dominance, or if the facilitator has created a situation of dominance by allocating more gaze attention to them, which contributed to the dominance scores assigned by the annotators.

Finally, a relationship is evident between ranked dominance group membership and player→away gaze, with R1 players gazing away for a longer duration on average than R2 players. The mean frequency of this gaze act is very similar between R1 and R2 groups and a high p-value indicates that the null hypothesis should be maintained i.e. there is no significant difference in the frequency at which R1 versus R2 players gaze away. On this basis, dominant players spend longer averting their gaze than their less dominant counterparts. This may be due to the fact that they also occupy more speaking time on average, with averted gaze being an indicator that an individual is not yet ready to cede their turn.

The null hypotheses maintained are that there is no evidence of a relationship between absolute dominance and any of the four gaze acts in question.

Furthermore, players look at the facilitator at similar rates, suggesting that, in a situation of asymmetry, participants will look at the most dominant individual *ex officio* at similar rates.

It seems slightly odd to say that R1 players spend more time gazing at their fellow players and gazing away than R2 players while managing to still gaze at the facilitator at similar levels, given that these 3 categories of gaze behaviour are mutually exclusive and exhaustive – no two behaviours may co-occur for a given participant for a given period, and all possible behaviour is described by these categories (Holle and Rein, 2015). However, these are only the inferences that can be drawn about the population with 95% confidence based on the given data. We cannot infer that, though the duration of these gaze acts is longer, their frequency is significantly lower, but it is possible that slight differences account for the apparent incompatibility of these conclusions.

Chapter 5

Conclusion and Future Work

One of the challenges in undertaking this research was the sheer number of approaches and tools available for many problems in computer vision, often with no clear leader in performance as this is so often contingent on the specific environment and use case in question. Consequently, a large degree of trial-and-error was required to get each of the phases of video analysis to an adequate level of performance – with some of these earlier attempts and subsequent improvements being described over the course of this paper. Furthermore, the range of metrics and factors to consider in analysing gaze behaviour means that those discussed here are by no means exhaustive. There are many opportunities for further work using this newly available data based on discussion of related work in Section 1.2, and a wealth of other literature not referenced here which opens up even more possible avenues.

More specifically, gaze as it relates to conversational role should be examined, including whether the increased facilitator attention for R1 participants is a result of their greater average speaking time. It would be interesting to see if results reflect the findings of Fukuhara and Nakano (2011). It might also be interesting to explore if these findings are replicated in multiparty conversations where all participants are on an equal footing i.e. no one has the elevated role of facilitator. Exploration of mutual gaze is an obvious next step in pursuing research on gaze in MULTISIMO, and potential uses of fine-grained annotations that include blinking data should also be considered.

Performance of the automatic annotator could perhaps be improved by incorporating the head posture data generated, either by using it as a factor

contributing to the process of classifying gaze direction or to flag potentially inaccurate classifications where a contradiction occurs. Alternatively, as gaze and head posture are two factors that are related but by no means dependent, head posture data could provide an additional tier of data for analysis; it would be interesting to investigate whether any relationship can be established between certain combinations of gaze and head posture behaviour, and conversational dominance. One may imagine a situation where the facilitator gazes towards the less dominant player, while keeping their head angled towards the more dominant player, for example.

Finally, a comparison of the performance of this heuristic-based system to one based on deep learning could prove to be interesting. There are a number of libraries providing pre-trained deep learning models that may be augmented with domain specific data (Zhang et al., 2015).

In summary, in the context of MULTISIMO, dominant players spend more time looking at their fellow player than their less dominant counterparts and avert their gaze for longer periods on average. Though both dominant and submissive players gaze at the facilitator at similar rates, the dominant individual receives longer gazes on average than their fellow player.

The use of an automatic system enabled the collection of data that would ordinarily have taken weeks of tedious labour for a human annotator. It is hoped that this system and the data it produces will prove to be valuable assets for the corpus, and facilitate a variety of future research in MULTISIMO.

Bibliography

Argyle, M. (1967). Eye-contact and direction of gaze. *The psychology of interpersonal behaviour. Harmondsworth, Penguin Books.*

Argyle, M. and Dean, J. (1965). Eye-contact, distance and affiliation. *Sociometry*, 28(3):289 – 304.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555 – 596.

Bee, N., Pollock, C., André, E., and Walker, M. (2010). Bossy or wimpy: expressing social dominance by combining gaze and linguistic behaviors. In *International Conference on Intelligent Virtual Agents*, pages 265 – 271. Springer.

Canu, S. (2019). Eye detection - gaze-controlled keyboard with python and opencv. https://pysource.com/2019/01/07/eye-detection-gaze-controlled-keyboard-with-python-and-opencv-p-1/.

Costello, R. (2018). Analysing dominance in multi-party dialogue. Bachelor's thesis, Trinity College Dublin.

Dawson-Howe, K. (2014). *A Practical Introduction to Computer Vision with OpenCV*. John Wiley & Sons.

Foddy, M. (1978). Patterns of gaze in cooperative and competitive negotiation. *Human relations*, 31(11):925 – 938.

Fukayama, A., Ohno, T., Mukawa, N., Sawaki, M., and Hagita, N. (2002). Messages embedded in gaze of interface agents - impression management with agent's gaze. In *Proceedings of the SIGCHI Conference: Human Factors in Computing Systems*, pages 41 – 48.

Fukuhara, Y. and Nakano, Y. (2011). Gaze and conversation dominance in multiparty interaction. In *2nd workshop on eye gaze in intelligent human machine interaction*, volume 9, pages 9 – 16.

Gobel, M., Chen, A., and Richardson, D. (2017). How different cultures look at faces depends on the interpersonal context. *Canadian Journal of Experimental Psychology*, 71(3):258 – 264.

Gobel, M. S., Kim, H. S., and Richardson, D. C. (2015). The dual function of social gaze. *Cognition*, 136:359 – 364.

Hessels, R. S., Holleman, G. A., Kingstone, A., Hooge, I. T., and Kemner, C. (2019). Gaze allocation in face-to-face communication is affected primarily by task structure and social context, not stimulus-driven factors. *Cognition*, 184:28 – 43.

Heylen, D., Van Es, I., Nijholt, A., and van Dijk, B. (2002). Experimenting with the gaze of a conversational agent. In *Proceedings international CLASS workshop on natural, intelligent and effective interaction in multimodal dialogue systems*, pages 93 – 100.

Holle, H. and Rein, R. (2015). Easydiag: A tool for easy determination of interrater agreement. *Behavior Research Methods*, 47(3):837.

Itakura, H. (2001). Describing conversational dominance. *Journal of Pragmatics: An Interdisciplinary Journal of Language Studies*, 33(12):1859 – 1880.

Kendon, A. (1967). Some functions of gaze-direction in social interaction. *Acta Psychologica*, 26(1):22 – 63.

Koutsombogera, M., Costello, R., and Vogel, C. (2018). Quantifying dominance in the multisimo corpus. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 141 – 146.

Koutsombogera, M. and Vogel, C. (2017). The multisimo multimodal corpus of collaborative interactions. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 502 – 503.

Libby, W. L. and Yaklevich, D. (1973). Personality determinants of eye contact and direction of gaze aversion. *Journal of Personality and Social Psychology*, 27(2):197 – 206.

Malik, U., Barange, M., Ghannad, N., Saunier, J., and Pauchet, A. (2019). A generic machine learning based approach for addressee detection in multiparty interaction. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pages 119 – 126.

Mallick, S. (2016). Head pose estimation using opencv and dlib. https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/.

McNeill, D. (2005). Gesture, gaze, and ground. In *International workshop on machine learning for multimodal interaction*, pages 1 – 14.

Norris, S. (2004). *Analyzing multimodal interaction: A methodological framework*. Routledge.

Sanchez-Cortes, D., Aran, O., Mast, M., and Gatica-Perez, D. (2012). A nonverbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia*, 14(3):816 – 832.

Tacchetti, M. (2018). *User Guide for ELAN Linguistic Annotator: Version 5.0.0*. The Language Archive, MPI for Psycholinguistics, Nijmegen, The Netherlands.

Vogel, C., Koutsombogera, M., and Costello, R. (2020). Analyzing likert scale inter-annotator disagreement. In *Neural Approaches to Dynamics of Signal Exchanges*, pages 383 – 393. Springer.

Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. (2015). Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511 – 4520.

Appendix A

Source Code

A.1 Python Script for Video Analysis

```
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import dlib
from math import hypot

video_name = 'short'
cap = cv2.VideoCapture(video_name + '.mov')
_, frame = cap.read()

detector = dlib.get_frontal_face_detector()
predictor =
    dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

FONT = cv2.FONT_HERSHEY_PLAIN


def midpoint(pt1, pt2):
    return int((pt1.x + pt2.x)/2), int((pt1.y + pt2.y)/2)
```

```
def get_blinking_ratio(facial_landmarks, eye_points):
    left_point =        (facial_landmarks.part(eye_points[0]).x,
                         facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x,
                         facial_landmarks.part(eye_points[3]).y)
    centre_top = midpoint(facial_landmarks.part(eye_points[1]),
                         facial_landmarks.part(eye_points[2]))
    centre_bottom = midpoint(facial_landmarks.part(eye_points[5]),
                         facial_landmarks.part(eye_points[4]))

    # CHECKING RATIO OF HORIZONTAL AND VERTICAL
     LINES USING EUCLIDEAN DISTANCE
    horizontal_line_length = hypot(left_point[0] - right_point[0],
                left_point[1] - right_point[1])
    vertical_line_length = hypot(centre_top[0] - centre_bottom[0],
                centre_top[1] - centre_bottom[1])

    # AVOIDING DIVISION BY ZERO ERROR
    if vertical_line_length != 0:
        ratio = horizontal_line_length / vertical_line_length
    else:
        ratio = 5

    return ratio


def get_gaze_ratio(facial_landmarks, eye_points):
    # ISOLATING EYE REGION
    eye_region = np.array([
                (facial_landmarks.part(eye_points[0]).x,
                facial_landmarks.part(eye_points[0]).y),
                 (facial_landmarks.part(eye_points[1]).x,
                 facial_landmarks.part(eye_points[1]).y),
                 (facial_landmarks.part(eye_points[2]).x,
                facial_landmarks.part(eye_points[2]).y),
                 (facial_landmarks.part(eye_points[3]).x,
                facial_landmarks.part(eye_points[3]).y),
```

```
                (facial_landmarks.part(eye_points[4]).x,
                facial_landmarks.part(eye_points[4]).y),
                (facial_landmarks.part(eye_points[5]).x,
                facial_landmarks.part(eye_points[5]).y)],
                np.int32)

eye = gray
min_x = np.min(eye_region[:, 0])
max_x = np.max(eye_region[:, 0])
min_y = np.min(eye_region[:, 1])
max_y = np.max(eye_region[:, 1])

# THRESHOLDING EYE REGION TO ISOLATE IRIS
if max_y - min_y > 0 and max_x - min_x > 0:
    gray_eye = eye[min_y:max_y, min_x:max_x]
    cv2.imshow("Eye", gray_eye)

    _, threshold_eye = cv2.threshold(gray_eye, 127, 255,
                    cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    threshold_eye = cv2.resize(threshold_eye,
                    None, fx=5, fy=5)
    threshold_eye = cv2.erode(threshold_eye,
                    None, iterations=2)
    threshold_eye = cv2.dilate(threshold_eye,
                    None, iterations=4)

    # DIVIDING THRESHOLDED REGION IN HALF TO ALLOW
            FOR GAZE DETECTION
    height, width = threshold_eye.shape

    left_side_threshold =
                threshold_eye[0:height, 0:int(width / 2)]
    right_side_threshold =
                threshold_eye[0:height, int(width / 2):width]

    # COUNTING NUMBER OF WHITE PIXELS IN EACH
    HALF TO DETERMINE WHICH CONTAINS MORE IRIS
    left_side_white = cv2.countNonZero(left_side_threshold)
```

```
            right_side_white = cv2.countNonZero(right_side_threshold)

            cv2.imshow("Threshold", threshold_eye)
            cv2.imshow("Left", left_side_threshold)
            cv2.imshow("Right", right_side_threshold)

            # AVOIDING ERROR IF EYE NOT FOUND (E.G. IF EYE CLOSED)
            if left_side_white == 0:
                gaze_ratio = 1
            elif right_side_white == 0:
                gaze_ratio = 3
            else:
                gaze_ratio = left_side_white / right_side_white
        else:
            gaze_ratio = 1.5

        return gaze_ratio


gaze_df = pd.DataFrame(columns=
        ['Frame', 'Direction', 'Rotation Vector','Timestamp'])
frame_count = 1

while(cap.isOpened()):
    ret, frame = cap.read()

    if ret == True:
        size = frame.shape
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = detector(gray)
        for face in faces:
            # DRAWING RECTANGLE AROUND FACE
            x, y = face.left(), face.top()
            x1, y1 = face.right(), face.bottom()
            cv2.rectangle(frame, (x, y),
                (x1, y1), (0, 255, 0), 1)

            # DETECTING BLINKING
```

```
landmarks = predictor(gray, face)
left_eye_ratio = get_blinking_ratio(
    landmarks, [36, 37, 38, 39, 40, 41])
right_eye_ratio = get_blinking_ratio(
    landmarks, [42, 43, 44, 45, 46, 47])
blinking_ratio = (left_eye_ratio
    + right_eye_ratio) / 2

# DETECTING GAZE DIRECTION
gaze_ratio_left = get_gaze_ratio(
    landmarks, [36, 37, 38, 39, 40, 41])
gaze_ratio_right = get_gaze_ratio(
    landmarks, [42, 43, 44, 45, 46, 47])
gaze_ratio = (gaze_ratio_left
    + gaze_ratio_right) / 2

# INDICATING GAZE DIRECTION
if blinking_ratio >= 5:
    cv2.putText(frame, "BLINKING", (50, 100),
                    FONT, 2, (0, 0, 255), 3)
    gaze_direction = "Blinking"
else:
    if gaze_ratio < 1:
        cv2.putText(frame, "LEFT", (50, 150),
            FONT, 2, (0, 0, 255), 3)
        gaze_direction = "Left"
    elif 1 < gaze_ratio < 2:
        cv2.putText(frame, "GAZE_AWAY", (
            50, 150), FONT, 2, (0, 0, 255), 3)
        gaze_direction = "Gaze_Away"
    else:
        cv2.putText(frame, "RIGHT", (50, 150),
            FONT, 2, (0, 0, 255), 3)
        gaze_direction = "Right"

#2D IMAGE POINTS
image_points = np.array([
            (landmarks.part(33).x,
```

```
                landmarks.part(33).y),  # Nose tip
                (landmarks.part(8).x,
                landmarks.part(8).y),   # Chin
                (landmarks.part(36).x,
                landmarks.part(36).y),  # Left eye corner
                (landmarks.part(45).x,
                landmarks.part(45).y),  # Right eye corner
                (landmarks.part(48).x,
                landmarks.part(48).y),  # Left mouth corner
                (landmarks.part(54).x,
                landmarks.part(54).y)   # Right mouth corner
            ], dtype="double")

    # 3D MODEL POINTS
    model_points = np.array([
                (0.0, 0.0, 0.0),             # Nose tip
                (0.0, -330.0, -65.0),        # Chin
                (-225.0, 170.0, -135.0),     # Left eye corner
                (225.0, 170.0, -135.0),      # Right eye corner
                (-150.0, -150.0, -125.0),    # Left mouth corner
                (150.0, -150.0, -125.0)      # Right mouth corner

                            ])

    # CAMERA INTERNALS
    focal_length = size[1]
    center = (size[1]/2, size[0]/2)
    camera_matrix = np.array(
                             [[focal_length, 0, center[0]],
                             [0, focal_length, center[1]],
                             [0, 0, 1]], dtype = "double"
                             )

    dist_coeffs = np.zeros((4,1))
    # Assuming no lens distortion
    (success, rotation_vector, translation_vector) =
        cv2.solvePnP(model_points, image_points,
        camera_matrix, dist_coeffs,
```

```
                    flags=cv2.SOLVEPNP_ITERATIVE)

                # PROJECTING LINE INDICATING
                HEAD ORIENTATION
                (nose_end_point2D, jacobian) =
                    cv2.projectPoints(np.array([(0.0, 0.0, 1000.0)]),
                    rotation_vector, translation_vector,
                    camera_matrix, dist_coeffs)

                p1 = ( int(image_points[0][0]),
                    int(image_points[0][1]))
                p2 = ( int(nose_end_point2D[0][0][0]),
                    int(nose_end_point2D[0][0][1]))

                cv2.line(frame, p1, p2, (255,0,0), 2)

                timestamp = cap.get(cv2.CAP_PROP_POS_MSEC)

                gaze_df = gaze_df.append(
                    {'Frame': frame_count, 'Direction': gaze_direction,
                    'Rotation Vector':
                      ','.join(str(x) for x in rotation_vector),
                    'Timestamp': timestamp}, ignore_index=True)
                frame_count += 1

                cv2.imshow("Frame", frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

    else:
        break

cap.release()
cv2.destroyAllWindows()

csv_name = video_name + "_df.csv"
gaze_df.to_csv(csv_name, index=False)
```

A.2 Python Script for Creating Fine Annotations

```
import pandas as pd
import numpy as np

video_name = 'short'
gaze_df = pd.read_csv(video_name + "_df.csv")

for index, row in gaze_df.iterrows():
    if row['Direction'] == 'Left':
        gaze_df.loc[index,'Direction'] = 'Gaze_Player-Left'
        # gaze_df.loc[index,'Direction'] = 'Gaze_Player'
        # gaze_df.loc[index,'Direction'] = 'Gaze_Away'
        # gaze_df.loc[index,'Direction'] = 'Gaze-Facilitator'
    elif row['Direction'] == 'Right':
        gaze_df.loc[index,'Direction'] = 'Gaze_Player-Right'
        # gaze_df.loc[index,'Direction'] = 'Gaze_Player'
        # gaze_df.loc[index,'Direction'] = 'Gaze-Facilitator'
    # elif row['Direction'] == 'Gaze_Away':
    #     gaze_df.loc[index,'Direction'] = 'Gaze_Player'

new_df = pd.DataFrame(columns=['Begin Time', 'End Time',
        'Direction', 'Duration'])
prev_dir = gaze_df.loc[1, 'Direction']
prev_timestamp = 0.0
for index, row in gaze_df.iterrows():
    if prev_dir != row['Direction']:
        curr_timestamp = row['Timestamp']
        duration = curr_timestamp - prev_timestamp
        new_df = new_df.append({'Begin Time': prev_timestamp,
                        'End Time': curr_timestamp,
                'Direction': prev_dir, 'Duration': duration},
                ignore_index=True)
        prev_timestamp = curr_timestamp
    prev_dir = row['Direction']
```

```
curr_timestamp = gaze_df['Timestamp'].iloc[-1]
duration = curr_timestamp - prev_timestamp
new_df = new_df.append({'Begin Time': prev_timestamp,
        'End Time': curr_timestamp,
        'Direction': prev_dir, 'Duration': duration},
        ignore_index=True)

# REMOVING ENTRIES <= 100ms UNLESS BLINKING
new_df = new_df[((new_df['Duration'] > 100)
        | (new_df['Direction'] == 'Blinking'))]
new_df.index = range(len(new_df))

for index,row in new_df.iterrows():
    if index >= 1 and
        new_df.loc[index-1,'End Time'] !=
                new_df.loc[index,'Begin Time']:
                        new_df.loc[index,'Begin Time'] =
                                new_df.loc[index-1,'End Time']

# MERGING NEIGHBOURING ROWS WITH SAME ANNOTATION VALUE
prev_dir = 'none'
rows_to_drop = []
for index,value in new_df['Direction'].items():
    if value == prev_dir:
        new_df.loc[index,'Begin Time'] =
                        new_df.loc[index-1,'Begin Time']
        new_df.loc[index,'Duration'] =
                        new_df.loc[index,'End Time']
                - new_df.loc[index,'Begin Time']
        rows_to_drop.append(index-1)
    prev_dir = value

new_df = new_df.drop(rows_to_drop)

# ROUNDING TO WHOLE TIME UNITS
new_df['Begin Time'] = np.floor(new_df['Begin Time'])
new_df['End Time'] = np.floor(new_df['End Time'])
new_df['Duration'] = np.floor(new_df['Duration'])
```

```
# CONVERTING TO INTS
new_df['Begin Time'] =
        pd.to_numeric(new_df['Begin Time'], downcast='signed')
new_df['End Time'] =
        pd.to_numeric(new_df['End Time'], downcast='signed')
new_df['Duration'] =
        pd.to_numeric(new_df['Duration'], downcast='signed')

annotation_name = video_name + "_annotation.csv"
new_df.to_csv(annotation_name, index=False)
```

A.3 Python Script for Creating Broad Annotations

```
import pandas as pd
import sys
from pathlib import Path

filename = sys.argv[1]
annotations = pd.read_csv(filename)

annotations =
        annotations.loc[annotations['Direction'] != 'Blinking']
annotations.index = range(len(annotations))

for index,row in annotations.iterrows():
    if index >= 1 and
        annotations.loc[index-1,'End Time'] !=
                annotations.loc[index,'Begin Time']:
                                annotations.loc[index,'Begin Time']
                                = annotations.loc[index-1,'End Time']
prev_dir = 'none'
rows_to_drop = []
for index,value in annotations['Direction'].items():
    if value == prev_dir:
        annotations.loc[index,'Begin Time'] =
                        annotations.loc[index-1,'Begin Time']
        annotations.loc[index,'Duration'] =
                        annotations.loc[index,'End Time']
                - annotations.loc[index,'Begin Time']
        rows_to_drop.append(index-1)
    prev_dir = value

annotations = annotations.drop(rows_to_drop)
name = Path(filename).stem + "_broad.csv"
annotations.to_csv(name, index=False)
```

Appendix B

Example Annotation File

An example of the type of CSV file produced after video analysis and annotation filtering is performed to produce broad annotations. Note that real annotation files are significantly longer.

Begin Time	End Time	Direction	Duration
0	133	Gaze-Facilitator	133
133	1701	Gaze_Player	1568
1701	3103	Gaze-Facilitator	1402
3103	3670	Gaze_Player	533
3670	14981	Gaze-Facilitator	11311
14981	16016	Gaze_Player	1035
16016	21388	Gaze-Facilitator	5271
21388	21621	Gaze_Away	233
21621	21855	Gaze_Player	233
21855	22389	Gaze_Away	533
22389	27694	Gaze-Facilitator	5305
27694	28261	Gaze_Away	567
28261	37404	Gaze-Facilitator	9143
37404	38605	Gaze_Player	1201
38605	39305	Gaze-Facilitator	700
39305	39472	Gaze_Player	166