Auftrag zur Auswertung der Eyetracking-Daten in Matlab

Ziel ist es, die Fixationen, die der Eyetracker während des Lesens misst, mit der Position der Wörter auf dem Bildschirm abzugleichen und davon eine Tabelle zu erstellen, die wie folgt aussieht:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **group** | **text** | **condition** | **time** | **sentence** | **type** | **word** | **duration** | **Avg.PS** |
| CE8 | TraMa | Text1 | SE | First | 1 | word |  | 154 |  |
| CE8 | TraMa | Text1 | SE | First | 1 | word | social | 230 |  |
| CE8 | TraMa | Text1 | SE | First | 1 |  |  |  |  |
| CE8 | TraMa | Text1 | SE | First | 1 |  |  |  |  |
| CE8 | TraMa | Text1 | SE | First | 1 |  |  |  |  |
| CE8 | TraMa | Text1 | SE | First | 1 |  |  |  |  |

Tipp: du musst die Tabelle zuerst als «Cell Matrix» vorbereiten, bevor du sie mittels «loops» mit «char arrays» und «doubles» füllen kannst. Evtl. musst du zuerst alle relevanten Infos sammeln, bevor du sie in eine Tabelle einfügst. Wichtig ist, dass die Bezüge auf Variablen allgemein gehalten sind, damit sie sich dann auf alle Files aller VPs verallgemeinern lassen. Bei Loops empfiehlt es sich, Laufvariablen zu nehmen, welche sinnverwandt sind (t für text, s für sentence, f für fixation usw.).

Wo Du die Info findest:

id: Fullresults\_ELF\_CE8.mat 🡪 par.subjectID

group: siehe unten 🡪 nützliche Codes

text, condition, time:

Der erste Trigger der Datei CE8\_E1T1\_SE\_ET.mat und Variable «event» entspricht dem Wert par.trigger.exp\_\*\_start. Dabei lässt sich text, condition und time herauslesen und es gilt:   
 101 🡪 T1\_SE, E1 (time 🡪 first)  
 102 🡪 T1\_ELF, E1 (time 🡪 first)  
 103 🡪 T2\_SE, E1 (time 🡪 first)  
 104 🡪 T2\_ELF, E1 (time 🡪 first)

109🡪 T1\_SE, E2 (time 🡪 second)  
 110 🡪 T1\_ELF, E2 (time 🡪 second)  
 111 🡪 T2\_SE, E2 (time 🡪 second)  
 112 🡪 T2\_ELF, E2 (time 🡪 second)

sentence:

Info, welcher Satz des Textes gemeint ist. In der Variable «event» entspricht jeder Trigger 21/22/23/24 dem Anfang eines neuen Satzes, der erste Trigger 21 entspricht also dem ersten Satz, der zweite Trigger dem zweiten Satz etc… Leider unterscheidet sich die Anzahl der Sätze zwischen den Texten (38,38,28,26), womit jedesmal überprüft werden muss, ob es sih noch um einen Satz-Trigger handelt. Wichtig: die grosse Zahl neben dem Satz-Trigger ist die Latenz in Millisekunden bei der Auslösung des Triggers. Entsprechend wurde dieser Satz in der Zeitspanne bis zum nächsten Trigger präsentiert (als Beispiel der erste Satz von 1763350 – 1796565, ensprechend 33215 ms oder 33.215 sek)

type:

Ob eine Fixation ein Wort war (word), kein Wort (unknown) oder eine Regression (ein Wort, welches schon einmal vorkam 🡪 regression). Du findest alle Fixationen in der Variablen eyeevent.fixatons.data. Die erste Spalte ist die Latenz, die zweite die Endtime. Zuerst musst du herausfinden, welche Fixationen innerhalb der Satzpräsentation waren (z.B. für den ersten Satz die Fixationen zwischen 1763350 und 1796565, entsprechend die Fixationen 86 bis 191, evtl kann mann das mit einem «while»-loop lösen, der die Aktionen ausführt, solange man innerhalb des Zeitrahmens ist). Dann gleichst du die X- (Spalte 4) und Y-Koordinate (Spalte 5) mit der Variablen wordbounds\_reading des entsprechenden Textes 1-4 und Satz ab. Diese Variable bezeichnet für jedes Wort eines Satzes in der ersten beiden Spalten die x- und y-Koordinate des Punktes oben links und die beiden nachfolgenden Spalten bezeichnen dies für den Punkt unten rechts. Also musst du z.b. beim Wort abgleichen, ob das X der Fixation zwischen den Werten X irgendeines Wortes liegt und entsprechend dasselbe auch für Y zutrifft. Beim Beispiel wäre die erste Fixation z.B. das Wort 13, die zweite Fixation das Wort 2, die dritte wiederum 2 (ich weiss noch nicht, ob dies schon als Regression gelten soll), die vierte 4 etc… Im Optimalfall baust du einen variablen Puffer von «z» Pixeln ein, innerhalb derer die Fixation immer noch als Wort gewertet wird (ich kenne die Datenqualität und die Literatur noch zu schlecht).

word:

Abgleich des Wortes mit der Variablen sentences\_ue. Zb. das 2. Wort des 1. Satzes des 1. Textes ist «social».

duration (der Fixation):

Ergibt sich aus der Differenz zwischen der 2. Spalte minus der 1. Spalte pro Fixation der Variablen eyeevent.fixations.data.

avg.ps:

Der Wert der 6. Spalte pro Fixation der Variablen eyeevent.fixations.data.

Nützliche Codes:

table(…, ‘VariableNames’, { … , … , … })

group.pilot = {'C00','C21','C22','C23', 'C24', 'C25', 'C26', 'CT','CXY'};

group.traba = {'CA0','CA1','CA2','CA3','CA4','CA5','CA6','CA7','CA8','CA9','CB0','CB1','CB2','CB3','CB4','CB5','CB6','CB7','CB8','CB9','CC0','CC1','CC2','CC3','CC4','CC5','CC6','CC7','CC8','CC9','CD0','CD1','CD2','CD3','CD4','CD5','CD6','CD7','CD8','CD9'};

group.trama = {'CE0','CE1','CE2','CE3','CE4','CE5','CE6','CE7','CE8','CE9','CF0','CF1','CF2','CF3','CF4','CF5','CF6','CF7','CF8','CF9','CG0','CG1','CG2','CG3','CG4','CG5','CG6','CG7','CG8','CG9','CH0','CH1','CH2','CH3','CH4','CH5','CH6','CH7','CH8','CH9'};

group.trapro = {'CI0','CI1','CI2','CI3','CI4','CI5','CI6','CI7','CI8','CI9','CJ0','CJ1','CJ2','CJ3','CJ4','CJ5','CJ6','CJ7','CJ8','CJ9','CK0','CK1','CK2','CK3','CK4','CK5','CK6','CK7','CK8','CK9','CL0','CL1','CL2','CL3','CL4','CL5','CL6','CL7','CL8','CL9'};

group.mulba = {'CM0','CM1','CM2','CM3','CM4','CM5','CM6','CM7','CM8','CM9','CN0','CN1','CN2','CN3','CN4','CN5','CN6','CN7','CN8','CN9','CO0','CO1','CO2','CO3','CO4','CO5','CO6','CO7','CO8','CO9','CP0','CP1','CP2','CP3','CP4','CP5','CP6','CP7','CP8','CP9'};

group.mulma = {'CQ0','CQ1','CQ2','CQ3','CQ4','CQ5','CQ6','CQ7','CQ8','CQ9','CR0','CR1','CR2','CR3','CR4','CR5','CR6','CR7','CR8','CR9','CS0','CS1','CS2','CS3','CS4','CS5','CS6','CS7','CS8','CS9','CT0','CT1','CT2','CT3','CT4','CT5','CT6','CT7','CT8','CT9'};

group.mulpro = {'CU0','CU1','CU2','CU3','CU4','CU5','CU6','CU7','CU8','CU9','CV0','CV1','CV2','CV3','CV4','CV5','CV6','CV7','CV8','CV9','CW0','CW1','CW2','CW3','CW4','CW5','CW6','CW7','CW8','CW9','CX0','CX1','CX2','CX3','CX4','CX5','CX6','CX7','CX8','CX9'};

for i = 1:length(subject)

if sum(strcmp(subject(1,i),group.pilot)) > 0

subject(2,i) = {'Pilot'};

elseif sum(strcmp(subject(1,i),group.traba)) > 0

subject(2,i) = {'TraBa'};

elseif sum(strcmp(subject(1,i),group.trama)) > 0

subject(2,i) = {'TraMa'};

elseif sum(strcmp(subject(1,i),group.trapro)) > 0

subject(2,i) = {'TraPro'};

elseif sum(strcmp(subject(1,i),group.mulba)) > 0

subject(2,i) = {'MulBa'};

elseif sum(strcmp(subject(1,i),group.mulma)) > 0

subject(2,i) = {'MulMa'};

elseif sum(strcmp(subject(1,i),group.mulpro)) > 0

subject(2,i) = {'MulPro'};

end

end

Von N. Langer:

Eye-tracking preprocessing and feature extraction The EyeLink 1000 tracker processes eye-position data, identifying saccades, fixations and blinks. Saccades are detected by the velocity and acceleration of the eye movements. Here, SR-research default system parameters have been used to define saccades: an acceleration threshold of 8000° per sec2, a velocity threshold of 30° per sec, and a deflection threshold of 0.1°. Fixations were defined as time periods without saccades. The dataset therefore consists of (x,y) gaze location entries for individual fixations (Fig. 3). Coordinates were given in pixels with respect to the monitor coordinates (the upper left corner of the screen was (0,0) and down/right was positive). We also provide raw sample data that can be used to validate fixation detection settings. Further, a blink can be regarded as a special case of a fixation, where the pupil diameter is either zero or outside a dynamically computed valid pupil, or the horizontal and vertical gaze positions are zero. For later analysis, only fixations within the boundaries of each displayed word have been extracted. On

the x-axis, the word boundaries were extended so that they were adjacent (Fig. 3). A Gaussian mixture model was trained on (y-axis) gaze data for each sentence to improve allocation of the eye fixations to the corresponding text lines. The number of text lines determined the number of Gaussians to be fitted within the model. Subsequently, each gaze data point was clustered to the matching gaussian and the data were realigned. As a result, each gaze data point is clearly assigned to a specific text line. Data points distinctly not associated with reading (minimum distance of 50 pixels to the text) were excluded. On the basis of a previous eye-tracking corpus5 we have extracted the following eye-tracking features

in MATLAB (code available in the data repository (Data Citation 1)): (I) gaze duration (GD), the sum of all fixations on the current word in the first-pass reading before the eye moves out of the word; (II) total reading time (TRT), the sum of all fixation durations on the current word, including regressions; (III) first fixation duration (FFD), the duration of the first fixation on the prevailing word; (IV) single fixation duration (SFD), the duration of the first and only fixation on the current word; and (V) go-past time (GPT), the sum of all fixations prior to progressing to the right of the current word, including regressions to previous words that originated from the current word. For each of these eye-tracking features we have additionally computed the pupil size. Furthermore, we have extracted the number of fixations and mean pupil size for each word and sentence. Fixations that were shorter than 100 ms were excluded from the analyses, because these are unlikely to

reflect fixations relevant for reading16, however, the raw eye-tracking data are available to assess further potential eye-tracking features.