GaRSIVis: Improving the Predicting of Self-Interruption during Reading using Gaze Data

Jan Pilzer
University of British Columbia
pilzer@cs.ubc.ca

Shareen Mahmud
University of British Columbia
shareenm@cs.ubc.ca

Vanessa Putnam
University of British Columbia
vputnam@cs.ubc.ca

Xinhong Liu
University of British Columbia
xinhliu@cs.ubc.ca

Tamara Munzner
University of British Columbia
tmm@cs.ubc.ca

Figure 1: Overview of the interface showing a list of readings with their fixation counts. Users can select a reading to clean the data or run a prediction on all readings to investigate the gaze pattern associated with a self interruption.

ABSTRACT

Gaze pattern data provides a promising opportunity to create a predictive model of self-interruption during reading that could support active interventions to keep a reader's attention at times when self-interruptions are predicted to occur. We present two systems designed to help analysts create and improve such a model. We present *GaRSIVis*, (Gaze Reading Self-Interruption Visualizer), that integrates a visualization front-end suitable for data cleansing and a prediction back-end that can be run repeatedly as the input data is iteratively improved. It allows analysts refining the predictive model to filter out unwanted parts of the gaze data that should not be used in the prediction. It relies on data gathered by *GaRSILogger*, which logs gaze data and activity associated with interruptions during on-screen reading. By integrating data cleansing and our prediction results in our visualization, we enable analysts using *GaRSIVis* to come up with a comprehensible way of understanding self-interruption from gaze related features.

ETVIS'18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5787-6/18/06.
https://doi.org/10.1145/3205929.3205933

CCS CONCEPTS

• **Human-centered computing → Visualization systems and tools**;

KEYWORDS

Eye tracking, log visualization, self-interruption, reading tasks.

ACM Reference Format:
Jan Pilzer, Shareen Mahmud, Vanessa Putnam, Xinhong Liu, and Tamara Munzner. 2018. GaRSIVis: Improving the Predicting of Self-Interruption during Reading using Gaze Data. In *ETVIS'18: 3rd Workshop on Eye Tracking and Visualization, June 14–17, 2018, Warsaw, Poland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3205929.3205933

1 INTRODUCTION

Reading activities such as reading a scientific paper or an article require the reader's attention for a significant period of time. In today's digital era, we often read papers on our computers and it is not uncommon to self-interrupt by going to social media or by simply losing focus and looking away. We define self-interruption as any activity that is initiated by the reader not related to reading.

We envision a reader application that has the ability to trigger adaptive intervention before a self-interruption is about to occur. In order to take steps towards this vision, we need to build a model that is able to predict self-interruptions. Given the established link between eye tracking and cognitive load, we believe eye tracking data can uncover valuable insights to be used as input for our predictor. For an accurate prediction, high quality input data is required. When raw data does not fit the expected quality, automated or manual data cleansing can transform noisy data into valid input.

To collect the required data, we built *GaRSILogger*, a reader application that connects to an eye tracker. We recorded eye tracking and self-interruption data of several participants using *GaRSILogger* to read a document of their choice. Our data is collected based on the notion that participants will not self-interrupt when they are in a focused lab setting. Therefore our data is collected in the setting most natural to the participants. They use their own device in a chosen environment. This setting does not allow us to record the self-interruptions of our participants by observation. We rely on the automatic logging we built into *GaRSILogger* based on our definition of self-interruption.

We developed *GaRSIVis* to show the participants' data and prediction results to analysts working on a predictor for self-interruptions. Our tool consists of a visualization front-end and a prediction back-end containing a fast predictor that can be run repeatedly at the discretion of the analyst with the push of a button. Raw eye-tracking data usually requires some cleaning. With our data, we found it necessary to look at the data quality in detail and trim certain parts manually. We included manual cleansing capabilities in *GaRSIVis* to enable analysts to refine unfamiliar data iteratively as a human in the loop. Having identified sections with missing or bad data, analysts can mark the selection as invalid and re-run the prediction.

We see our current model as a preliminary step towards creating a more reliable predictor for self-interruptions. We recognize our data cleansing functionality as an instrument to improve input data quality thereby improving the accuracy of our model.

2 RELATED WORK

Previous studies show that people can experience interruption every four to eight minutes, and about half of them can be self-interruptions [Czerwinski et al. 2004; González and Mark 2004]. Interruptions during work decrease productivity and affect people's sense of accomplishment [Murphy 2016]. Specifically, self-interruptions have been shown to be more disruptive than external interruptions [Katidioti et al. 2016].

Work related to eye tracking has grown over the last decade. Eye movement data that shows fixations and saccades can be used to estimate a person's cognitive load [Wang et al. 2014]. Tsai et al. used fixation durations to examine attention and found that an increased cognitive load is likely to cause distraction [Lavie 2010; Tsai et al. 2012]. For this reason, we hypothesize that when readers are cognitively weighed down, they tend to trigger self-interruptions. This prior work only suggests that gaze pattern can be used to indicate distraction but does not propose any visual representation directly connecting the two.

The importance of visualization for a successful analysis of eye tracking data has been confirmed in several different contexts [Sharif et al. 2012; Uwano et al. 2006]. Blascheck et al. show an overview of the existing eye tracking visualization approaches collected from various literatures [Blascheck et al. 2014]. However, these visualizations are typically static, support little interaction, and focus on areas of interest (AOI) that participants look at. In addition, Jo et al. propose a bookmarking tool to help readers resume a reading activity by highlighting the word that the reader last fixated on before getting distracted [Jo et al. 2015]. No prior work has looked for universal patterns associated with self-interruption that apply independent of specific areas of interest.

3 LOGGING

We gathered eye tracking and self-interruption data by letting participants read research papers of their choice using our logging application, *GaRSILogger*.

3.1 Data

The *GaRSILogger* log file contains events timestamped at millisecond precision. We grouped fixation events into fixations and calculate saccades. We derived fixation count, fixation duration, saccade length, saccade duration, and saccade angle from these fixations and saccades.

The log also contains events marking a self-interruption. Based on these events we split each session into segments labeled as one of three categories: normal reading, reading before an interruption, and invalid. Time after an interruption is invalid until the next fixation inside the *GaRSILogger* application.

Sessions are divided into equal chunks and the derived gaze features are used as input to a predictor to be classified as a normal reading or reading before an interruption. Each of these chunks has a ground truth label, a predicted label, and a flag if it was correctly classified.

3.2 Implementation

Our logger, *GaRSILogger*, is an operating system independent PDF viewer application giving participants a user experience comparable with existing viewer applications. The application connects to a Tobii Eye Tracker 4C to capture the participants' fixations and tracks window switches using information provided by the operation system. Window switches that last longer than 10 seconds are considered self-interruptions. Anything below that threshold could originate from a notification pop-up or an accidental key press and is ignored.

After one minute without a participants' gaze in the *GaRSILogger* window, the participant is asked for a reason providing the choice between "external interruption" and "self-interruption". Only instance where the second option is chosen are considered for further processing. Instances of external interruptions are excluded because they are not guaranteed to present the same characteristics as self-interruptions and would decrease the precision of a prediction of self-interruptions.

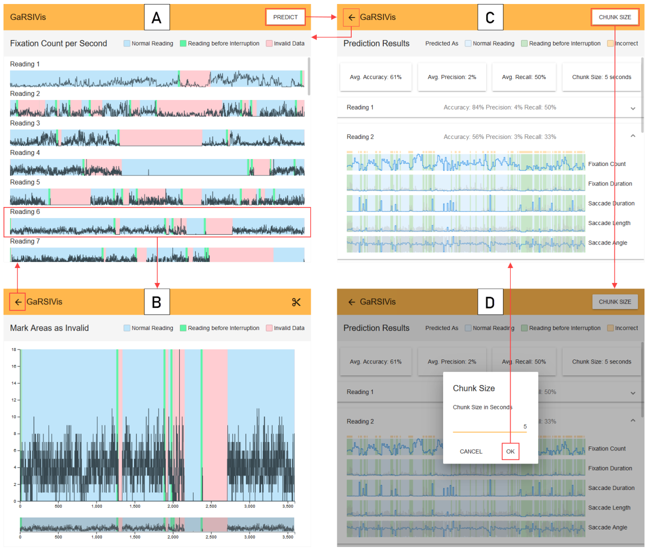

Figure 2: The Overview page (A) with the readings from which analysts can select a particular reading for cleaning. This reading is shown in the Data Cleansing page (B). Upon returning to the Overview page analysts can run a prediction on the cleaned data and view the Prediction Results page (C). From there analysts can choose to rechunk the data (D) and then rerun the prediction on the new chunk size. This workflow can be repeated.

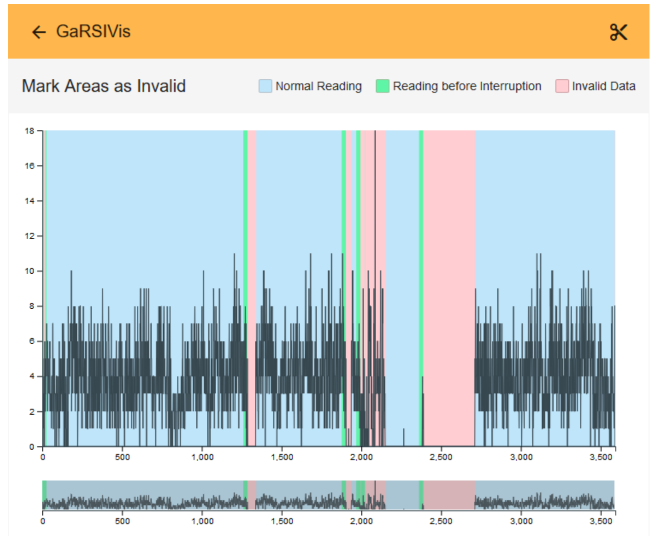

Figure 3: Data Cleansing: brush and zoom to inspect data in detail and trim invalid portions.

4 VISUALIZATION

We propose *GaRSIVis* to view, clean, and analyze our gathered data.

4.1 Interface

The *GaRSIVis* interface consists of three separate pages: Overview, Data Cleansing, and Prediction Results. The workflow of how analysts navigate between these pages is shown in Figure 2.

4.1.1 Data Overview. We use sparklines, condensed line charts without axes or coordinates, to record fixation count over time (see Figure 1). We display fixation count in the Overview since it is the best metric for determining if a portion of data is invalid or not. If the fixation count is zero for a length of time we can conclude that this piece of our data is invalid.

We choose color to encode our three categorical attributes: normal reading (blue), reading before an interruption (green), and invalid (red). For these colors we use light saturation to encode the background of each sparkline. Using red and green can be problematic for accessibility, but there is sufficient luminance difference between the colors to be distinguishable.

4.1.2 Data Cleansing. To analyze and tag invalid data segments, we have a navigation component from the list of readings to a page where a single reading can be analyzed in more detail (see Figure 3). For this we opt for a different view since when cleaning an analyst would only be interested in one specific area, rather than an overview of all data. We utilize the brush and zoom feature for this detailed inspection and manual cleaning annotation.

Our brush and zoom feature carries over the same sparklines and color encoding from the Data Overview. This page has a "context" view (bottom), containing the entire reading over time with encodings, as well as a "focus" view (top), a larger display showing the selected region in a zoomed-in detailed view. There are two user interactions that can allow an analyst to zoom in on the reading. First by brushing from the "context" view, and second by zooming in on the "focus" view.

To cleanse data an analyst selects an unwanted area using either of the options mentioned above. Next, by clicking the cut button in the top right corner the analyst marks an unwanted area as invalid, and excludes this portion of the data from further analysis.

4.1.3 Prediction Results. The Prediction Results page supports the detailed analysis of the features involved in the prediction (see Figure 4). Invalid data chunks are not shown since they are not used by the predictor. Summary statistics for accuracy, precision, and recall across all reading sessions are shown at the top of the page, as is the current chunk size. There is not enough room to show all of the features for all of the sessions, so each reading session can be collapsed or expanded. When collapsed, only the accuracy, precision, and recall statistics are shown. When the session is expanded, there is also a stepline strip for each of the gaze features. Steplines are sparklines, displaying discretized data instead of a continuum.

Each step corresponds to one chunk of aggregated data. The chunks are based on the timesize value t with a default of 5 seconds. By selecting "chunk size" the analyst can enter a new value to rerun the prediction with chunks of the new size. As the number of chunks before a self-interruption are limited, using a larger value for the length of a chunk changes the ratio of normal reading to reading before an interruption. When the value gets too large however, a chunk tagged as before an interruption contains data that should be considered normal reading.

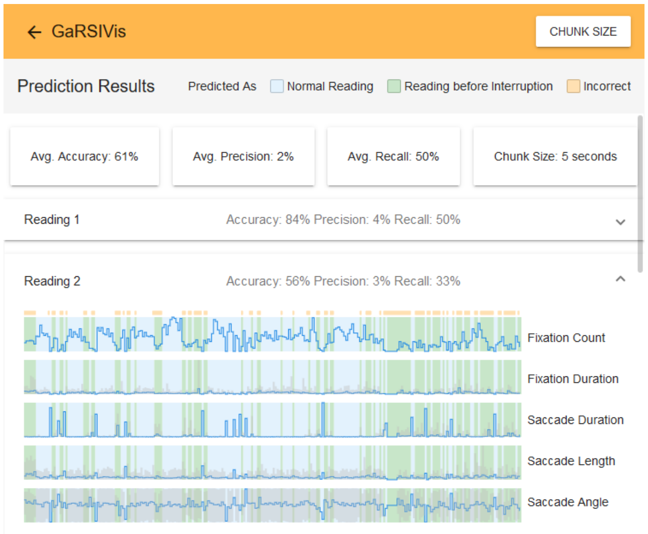

Figure 4: Prediction Results: examine classified results from gaze related features for normal readings and readings before an interruption.

The color coding in the Prediction Results page is consistent with the other two views: we use blue for normal reading chunks and green for chunks before an interruption. In addition, misclassified time chunks are tagged by small tick marks in gold. Invalid data is filtered out in this view, so there is no use of red.

4.2 Implementation

This visualization was implemented following a traditional client-server-architecture. The raw data was preprocessed in multiple steps before being used in the visualization.

The raw data is in the format of events in a log file. We group fixation events into fixations and calculate saccades. An initial chunking and prediction is performed with a default value for t. All preprocessing and prediction steps are done using Python and its prediction library scikit-learn [Pedregosa et al. 2011].

For our prediction, we use a Logistic Regression classifier. This classifier is simple to configure and fast enough to be run interactively without a long waiting time for the analyst. We perform k-fold cross-validation across participants using one session as test set each time. We chose this type of validation to judge a predictor on its performance to work across participants.

The server uses WebSockets for bidirectional communication enabling the server to run potentially lengthy calculations without concern for a timeout of the HTTP request. The server receives messages with manually marked invalid time sections and new chunk sizes, and asynchronously performs the required computation steps sending push notifications once results are available.

The visualization is implemented as a web application using Angular and following the Material Design guidelines. The sparklines and steplines are rendered as SVG using D3.js [Bostock et al. 2011].

5 DISCUSSION AND FUTURE WORK

The *GaRSIVis* interface helped us work towards predicting self-interruption by providing detailed views of a participant's gaze features. A long-term goal for this prediction is designing interventions to prevent self-interruption from happening in order to increase readers' productivity.

The strength of our work lies in including data cleansing and prediction result analysis in one interactive application. The predicted labels are shown against the gaze features used as input to the predictor allowing for better understanding of feature values associated with self-interruption. We not only show analysts the prediction results, but also keep the analyst in the loop by letting them examine the data quality and run the prediction on what they believe is “good” data. We believe this enhances the credibility of our system.

Our predictor shows the path towards finding a model that predicts self-interruptions. Future work could focus on improving the model, by using the insights gathered by our visualization. We would consider feature selection and evaluating more advanced classification approaches.

Additionally, even though we believe that the manual data cleaning component is a strength of our system, a certain degree of automatic cleaning needs to be added so that analysts do not feel overwhelmed. In future work, we would be interested to incorporate more intelligent automated cleaning to reduce the amount of manual annotation.

6 CONCLUSIONS

We propose *GaRSIVis*, an interactive visualization for eye tracking data towards understanding the gaze pattern related to self-interruption. We collected gaze data during reading activities alongside tagged self-interruptions using *GaRSILogger*. This collected data was used as an input to a preliminary prediction. The visualization of the prediction results allows analysts to analyze gaze related features associated with self-interruption. To improve data quality and therefore the prediction, we also provide interactive cleansing capabilities that allow analysts to mark unwanted areas of the data as invalid. Hence, *GaRSIVis* offers a comprehensible way of understanding self-interruption from gaze related features.

ACKNOWLEDGMENTS

We wish to thank Thomas Fritz for his guidance throughout this work. This work is supported in part by the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC.

REFERENCES

Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2014. State-of-the-Art of Visualization for Eye Tracking Data. In *EuroVis - STARs*. The Eurographics Association. https://doi.org/10.2312/eurovisstar.20141173

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 175–182. https://doi.org/10.1145/985692.985715

Victor M. González and Gloria Mark. 2004. "Constant, Constant, Multi-tasking Craziness": Managing Multiple Working Spheres. In *Proceedings of the SIGCHI Conference* *on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 113–120. https://doi.org/10.1145/985692.985707

Jaemin Jo, Bohyoung Kim, and Jinwook Seo. 2015. EyeBookmark: Assisting Recovery from Interruption During Reading. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2963–2966. https://doi.org/10.1145/2702123.2702340

Ioanna Katidioti, Jelmer P. Borst, Marieke K. van Vugt, and Niels A. Taatgen. 2016. Interrupt me: External interruptions are less disruptive than self-interruptions. *Computers in Human Behavior* 63 (Oct. 2016), 906–915. https://doi.org/10.1016/j.chb.2016.06.037

Nilli Lavie. 2010. Attention, Distraction, and Cognitive Control Under Load. *Current Directions in Psychological Science* 19, 3 (2010), 143–148. https://doi.org/10.1177/0963721410370295

Mark Murphy. 2016. Interruptions At Work Are Killing Your Productivity. *Forbes* (Oct. 2016).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830.

Bonita Sharif, Michael Falcone, and Jonathan I. Maletic. 2012. An Eye-tracking Study on the Role of Scan Time in Finding Source Code Defects. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 381–384. https://doi.org/10.1145/2168556.2168642

Meng-Jung Tsai, Huei-Tse Hou, Meng-Lung Lai, Wan-Yi Liu, and Fang-Ying Yang. 2012. Visual attention for solving multiple-choice science problem: An eye-tracking analysis. *Computers & Education* 58, 1 (Jan. 2012), 375–385. https://doi.org/10.1016/j.compedu.2011.07.012

Hidetake Uwano, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto. 2006. Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications (ETRA '06)*. ACM, New York, NY, USA, 133–140. https://doi.org/10.1145/1117309.1117357

Qiuzhen Wang, Sa Yang, Manlu Liu, Zike Cao, and Qingguo Ma. 2014. An eye-tracking study of website complexity from cognitive load perspective. *Decision Support Systems* 62 (June 2014), 1–10. https://doi.org/10.1016/j.dss.2014.02.007