EyeLiveMetrics: Real-time Analysis of Online Reading with Eye Tracking

Daniel Hienert
daniel.hienert@gesis.org
GESIS - Leibniz Institute for the Social Sciences
Cologne, Germany

Heiko Schmidt
heiko.schmidt@gesis.org
GESIS - Leibniz Institute for the Social Sciences
Cologne, Germany

Thomas Krämer
thomas.kraemer@gesis.org
GESIS - Leibniz Institute for the Social Sciences
Cologne, Germany

Dagmar Kern
dagmar.kern@gesis.org
GESIS - Leibniz Institute for the Social Sciences
Cologne, Germany

ABSTRACT

Existing eye tracking software have certain limitations, especially with respect to monitoring reading online: (1) Most eye tracking software record eye tracking data as raw coordinates and stimuli as screen images/videos, but without inherent links between both. Analysts must draw areas of interest (AOIs) on webpage text for more fine-grained reading analysis. (2) The computation and analysis of fixation and reading metrics are done after the experiment and thus cannot be used for live applications.

We present EyeLiveMetrics, a browser plugin that automatically maps raw gaze coordinates to text in real time. The plugin instantly calculates, stores, and provides fixation, saccade, and reading measures on words and paragraphs so that gaze behavior can be analyzed immediately. We also discuss the results of a comparative evaluation. EyeLiveMetrics offers a flexible way to measure reading on the web - for research experiments and live applications.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**; • **Information systems → Web applications**.

KEYWORDS

Online Reading, Reading Metrics, Real-time Analysis, Eye Tracking, Browser Plugin

1 INTRODUCTION

Eye tracking is a well-established method for measuring and analyzing the eye movements of individuals in various domains, such as marketing [Wedel and Pieters 2017], human-computer interaction (HCI) and usability [Poole and Ball 2006] as well as reading research [Just and Carpenter 1980]. This is partly because eye movements are good indicators of the user's attention, and a lot can be inferred about the user's cognitive state due to the correlation between eye movements and cognitive processes [Rayner 1998]. This makes eye tracking beneficial in experimental user studies to research human behavior. However, most eye tracking systems lack the generic functionality that maps raw eye tracking data to the visible stimuli. The mapping is usually done in the time-consuming process of drawing areas of interest (AOIs) by hand on each relevant screenshot. With the increasing interest in using eye tracking for reading-related activities on the web such as information retrieval [Buscher et al. 2012], online news consumption [Bhattacharya et al. 2020], fake news detection [Bozkir et al. 2022] or social media usage [Emily Vraga and Troller-Renfree 2016] this workflow is not efficient. Drawing AOIs for hundreds of web pages, each with multiple AOIs visited by several participants, is a major time commitment.

With the EyeMetricsLive plugin, we want to provide a solution for web stimuli in the form of a browser plugin that (1) captures eye tracking data live, (2) maps them to text on a web page even down to the word level, (3) calculates eye tracking metrics instantly during recording. The plugin enables researchers to analyze gaze data collected in experimental user studies efficiently and developers to adapt their interfaces based on previous gaze behavior, similar to the approach introduced as "The text 2.0 framework" by Bieder et al. [Biedert et al. 2010b]. For example, a web page can respond to reading comprehension difficulties by automatically switching to easy language, or a search engine can suggest search terms or re-rank search results based on what the user has read so far. Furthermore, the plugin eliminates the need for the time-consuming manual drawing of AOIs in eye tracking experiments, as it automatically maps gaze data to HTML elements. EyeLiveMetrics is an open-source modular plugin compatible with any eye tracker as long as it has access to the eye tracking data stream.

In this paper, we introduce EyeLiveMetrics, its processes and the available metrics. Furthermore, we give insights into the validity of EyeLiveMetrics.

2 RELATED WORK

Eye tracking has a long tradition in reading research and is considered a reliable indicator for individuals' moment-by-moment cognitive processing [Just and Carpenter 1980; Rayner 1998]. Various reading measures have been devised to capture these processes mainly based on fixation and saccades measures, like first-pass fixation duration, first-pass regression or re-reading duration. A common application of these reading measures is to classify whether a reader is reading a text in depth or skimming [Bhattacharya et al. 2020; Biedert et al. 2012; Gwizdka 2014; Kelton et al. 2019]. Although several studies have demonstrated the feasibility of this classification task, so far, no system or prototype has been developed to perform it in real time. Another promising application is the detection of reading comprehension, e.g., [Ahn et al. 2020; Just and Carpenter 2018; Mézière et al. 2023] in general and, more specifically, in the field of fake news detection on the web. Research by

Bozkir et al. [Bozkir et al. 2022] and Sümer et al. [Ömer Sümer et al. 2021] has shown that different eye movements are associated with reading news articles with true and false information.

In the context of live applications, Biedert et al. introduced the Text 2.0 framework [Biedert et al. 2010b], which allows us to react to reading behavior in real time. In the Eye Book [Biedert et al. 2010a], the user's gaze behavior is used to determine which text segments are currently being read, and based on this, additional effects such as sounds or images are generated and presented to the user. Another real-time approach in Interactive Information Retrieval uses gaze data to suggest query reformulations in a search engine prototype [Eickhoff et al. 2015]. In both examples, the system knows what the user is currently reading or has read before.

Over the years, several software tools have been developed to support researchers in conducting eye tracking experiments and analyze gaze data. Some commercial tools, such as Tobii Pro Lab, support reading measures[1] for specific text created by the experimenter with an integrated editor. The tools then automatically generate AOIs for characters, words, and sentences. However, for other types of stimuli, such as web studies, analysts still need to draw AOIs manually. EyeMap [Tang et al. 2012] and the Reading-Protocol [Hienert et al. 2019] are open-source tools that overcome this drawback by mapping the gaze data to automatically generated AOIs based on HTML elements of the web page. Based on them, research prototypes have been built, such as WebgazeAnalyzer [Beymer and Russell 2005], WebEyeMapper [Reeder et al. 2001] and ReMA [Valdunciel et al. 2022], but none of them considers real-time usage. Webgazer.js [Papoutsaki et al. 2016] allows the integration of eye movement data into a live system. However, it doesn't take into account the underlying content. It estimates eye tracking data based on webcam input and current interaction behavior on the screen, which comes with a loss of accuracy. Therefore, it is not yet applicable to use cases where high accuracy is necessary. This is where the EyeLiveMetrics plugin comes in. It combines highly accurate, real-time eye tracking with automatic AOI detection.

3 EYELIVEMETRICS - THE BROWSER PLUGIN

In this section, we present the implementation of the open-source browser plugin EyeLiveMetrics[2]. The main goals of this plugin are (1) eliminating the time-consuming process of mapping eye tracking data to text with AOIs, (2) computing fixation-, saccades- or reading measures for text on web pages on the fly, (3) enabling real-time analysis of reading and therewith immediate reaction to detected gaze behavior in an interactive setting. In the following, we give an overview of how the plugin collects the eye tracking data and the data processing procedure, which includes the mapping from gaze data to the text of the web page and the computation of the eye tracking metrics.

3.1 Gaze Data Collection

EyeLiveMetrics is built upon the Reading Protocol tool [Hienert et al. 2019], which uses raw eye tracking data and stored HTML pages to compute fixations on words in a rendering process after the user experiment. Thereby, coordinates were mapped to the bounding boxes of individual words, representing the AOIs, and later stored as word-eye-fixations in a database. This process can be error-prone for experiments on the web with arbitrary web pages. The Reading Protocol only stores HTML pages once loaded, but many web pages are dynamic, reloading content via AJAX calls or having dynamic content items such as menus, drop-down boxes, or carousels. Using a web page that is only stored once after the initial loading for the mapping process would not reflect the state over the entire time frame that a subject reads it. Accordingly, this can lead to errors in the mapping process from eye tracking coordinates to AOIs.

EyeLiveMetrics performs the mapping process in real time. Figure 1 gives an overview of the flow of gaze data. We implemented a Python script to transfer eye tracking coordinates to the plugin. The script accesses the eye tracker with, in our case, the Tobii Pro SDK for Python[3]. The SDK provides a continuous stream of the two-dimensional display coordinates (x,y) with timestamps and the 3D space coordinates (x,y,z), gaze-origin and gaze-position in the user coordinate system. The Python script also acts as a WebSocket server and sends the coordinates to the plugin, which acts as a client.

The plugin itself is implemented as a Tampermonkey[4] user script. Tampermonkey is a plugin container that allows the execution of arbitrary JavaScript code on any web page. It supports many browsers, such as Chrome, Edge, Safari, Opera, or Firefox. It allows a pattern-based inclusion or exclusion of web pages for which the user script is executed.

3.2 Data Processing Procedure

Figure 2 shows the data processing procedure in the EyeLiveMetrics plugin. Eye tracking coordinates are received in the WebSocket client. It uses a Velocity-Threshold Identification (I-VT) algorithm [Salvucci and Goldberg 2000] to classify raw gaze points as fixations or saccades based on the eye's directional shift velocity. Input parameters are gaze-origin and gaze-position coordinates to compute the gaze vector and compare it with previous ones. It returns the classification of fixation or saccade based on an adjustable threshold, e.g., 30° per second. For gaze points classified as a saccade, we store their position and velocity. If classification changes to fixation, the final saccade is computed and mapped to an AOI. AOIs used for saccade mapping are text paragraphs, images, and videos. If the start- and endpoint lies within an AOI, the saccade is saved for this AOI. At last, we compute the saccade duration and length. In Section 3.3, we will describe the saccade metrics in more detail.

When the classifier identifies a fixation, a fixation object is created. It includes properties such as start- and end time, the mapped HTML element (word, image, video) as AOI, the object bounding box, and object-specific metrics (e.g., word index/offset in the text node). When the fixation ends, fixation and reading metrics are computed. We refer to the next Section 3.3 for a detailed overview. All data is stored locally in the browser and then every five seconds or when the status of the browser tab changes in a database, e.g., on tab close or change. In general, we store participant ID, stimulus

[1]https://connect.tobii.com/s/article/How-reading-metrics-work?language=en_US
[2]EyeLiveMetrics is an open-source software under GPLv3 licence at https://git.gesis.org/iir/eyelivemetrics
[3]https://developer.tobiipro.com/python/python-getting-started.html
[4]https://www.tampermonkey.net/

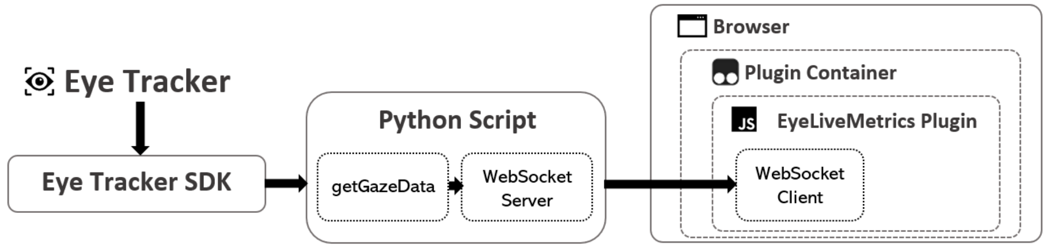

Figure 1: Flow of gaze data from eye tracker to the EyeLiveMetrics plugin

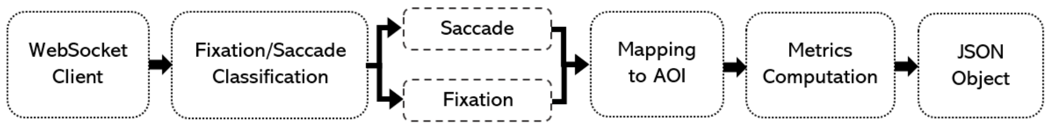

Figure 2: Data processing in the EyeLiveMetrics plugin

(URL of the web page), start-, end time, fixations, and saccades objects for words, images, and videos as well as the whole web page as text, a word and a sentence index for the whole web page text.

3.3 Metrics for Reading

The metrics of EyeLiveMetrics follow the implementation of fixation–, saccade-, and reading metrics by Tobii Pro Lab[5]. A full list of all available metrics can be found in the GIT repository[2] of EyeLiveMetrics.

Fixations are the prolonged retention of the gaze on one point, for textual stimuli on words to understand their syntactic and semantic meaning [Reichle et al. 2012]. EyeLiveMetrics stores fixation measures for each web page element, such as words. First, we have the common eye tracking metrics such as total fixation duration, number of fixations, timestamp of the first fixation, time to first fixation, first fixation duration, and average fixation duration. Further, we have measures specific to words, e.g., character index and sentence number in HTML documents. At last, we also have timestamps, fixation group numbers, and stimuli bounding boxes.

Saccades are rapid movements of the gaze between fixations. For text stimuli, they describe the movement from word to word. This means that saccades can abstractly describe the reading behavior on text. For example, to understand the scan path on paragraphs [Valdunciel et al. 2022] or for the distinction between reading and skimming [Biedert et al. 2012]. EyeLiveMetrics stores saccades for textual paragraphs. For each saccade, it stores the timestamps of the entry- and exit saccade, its sequential index, its duration, its length, its amplitude, its peak velocity, and its 2-dimensional direction vector.

Reading measures can describe additional characteristics of reading behavior, such as analyzing reading comprehension [Mézière et al. 2023]. In case the HTML element is a word, EyeLifeMetrics stores the following reading measures: First-pass first fixation duration, first-pass fixation group number, first-pass regression, first-pass duration, regression path duration, selective regression path duration, and re-reading duration.

[5]https://connect.tobii.com/s/article/understanding-tobii-pro-lab-eye-tracking-metrics?language=en_US

4 USAGE OF EYELIVEMETRICS

Once the EyeLiveMetrics plugin has been added to the browser and the other necessary resources (e.g. database and web server) have been set up according to the instructions on the project website, the setup is ready to be used either in a user study or in an interactive environment. In both cases, the eye tracker must be calibrated to the user's eyes using the calibration tool usually available in the eye tracker management software. Next, the browser can be opened, and the EyeLiveMetrics plugin can be activated. From now on, gaze data collection (see 3.1) will take place for each web page visited.

After the data collection in user studies, the experimenter can immediately see the analysis results by simply launching the analysis user interface (see Figure 3). This also allows the data to be used directly in retrospective interviews. Fixation-, saccades-, and reading metrics can be seen in a pop-up window for each word.

In real-time applications, the analyzed data can be used immediately to manipulate the interface. For example, Figure 4 shows how the intensity of reading a word can be shown directly to the reader. Here, the word's fixation duration determines its background color. Although the usefulness of this approach is questionable, it illustrates that it is possible to generate a heat map on the fly.

5 EVALUATION

We evaluated EyeLiveMetrics in two different ways. First, we examined the performance of EyeLiveMetrics and then compared the results with those of Tobii Pro Lab for the same text stimulus. We begin this section with a description of the technical environment we used in the two experiments and present both evaluations afterward.

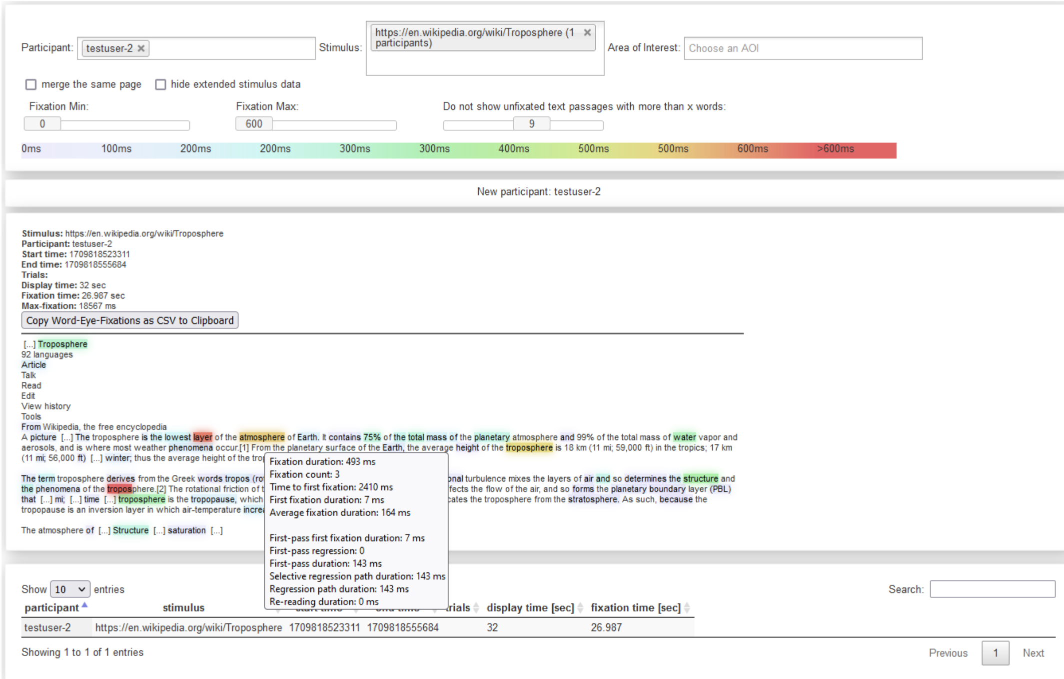

Figure 3: EyeLiveMetric's user interface showing analysis results adapted from the reading protocol UI [Hienert et al. 2019].

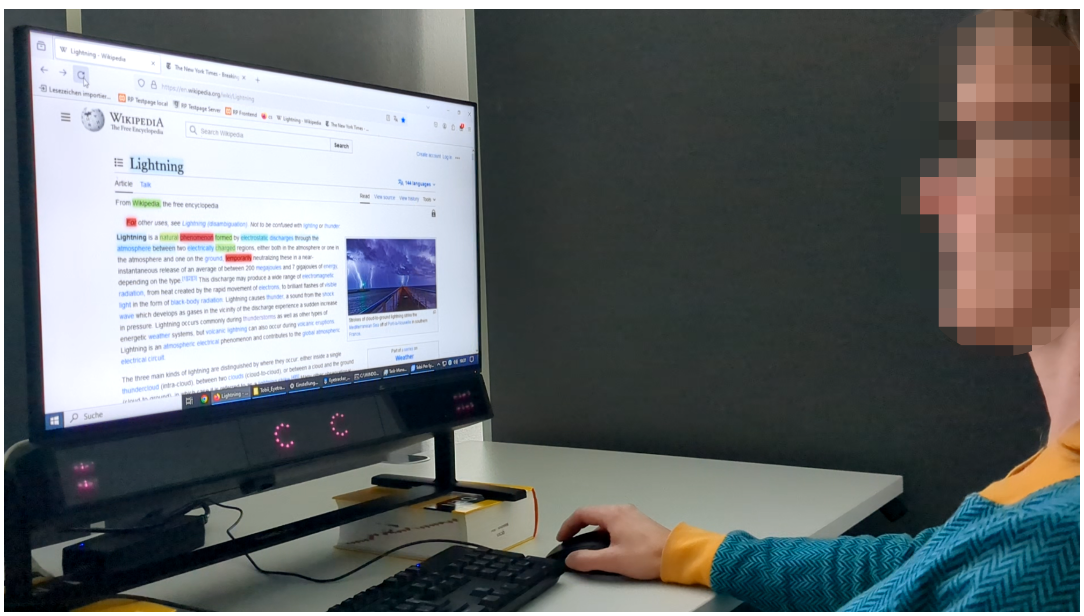

Figure 4: Participant reading a Wikipedia article and generating a heat map on the fly.

5.1 Technical evaluation environment

For the evaluation, we used a Tobii Pro Spectrum device with a sampling rate of 300 Hz. The sampling rate is an upper limit of how much time a live application such as EyeLiveMetrics can use to fully process a single gaze coordinate. Both experiments were conducted on a PC with Intel i7-6700 CPU@3.40GHz and 16GB RAM with Firefox V117.0 as a web browser. As mentioned in Section 3.1, we integrate EyeLiveMetrics as a Tampermonkey plugin.

Table 1: Comparison of eye tracking metrics of Tobii Pro Lab and EyeLiveMetrics. All time measures are in ms.

Metrics	Fixation Metrics						
	TFD	AFD	MiFD	MaFD	#F	TFF	FFD
Pearson	1.0	0.99	0.996	0.962	0.999	1.0	0.973
MAE + std	5.364 ± 6.067	0.734 ± 0.675	0.281 ± 0.119	3.879 ± 7.542	1.0 ± 1.128	3.595 ± 2.297	0.636 ± 1.327
	Reading Metrics						
	FpFFD	FpD	FpR	RPD	sRPD	RRD	
Pearson	0.979	1.0	1.0	1.0	1.0	1.0	
MAE + std	0.495 ± 1.226	3.757 ± 6.006	0.0 ± 0.0	4.393 ± 6.446	4.353 ± 5.884	1.021 ± 3.24	

5.2 Experiment: EyeLiveMetrics Performance

Since the EyeLiveMetrics is a browser plugin that computes eye tracking metrics in real time (cf. Section 3.3), its main loop has to run at least as fast as the eye tracker records and outputs raw gaze coordinates according to its sampling rate. For each coordinate, the plugin has to receive the coordinates, classify them into saccades or fixations, then map them to an HTML element (text, word, image, video), and at last, compute the fixation-, saccade- and reading metrics. Therefore, each iteration of the plugin has to be quick enough to handle the next raw gaze data point.

We conducted a performance test with the technical specifications described in Section 5.1 and recorded ten minutes of reading a large article on Wikipedia. The plugin recorded and processed n=176,091 raw gaze coordinates with an average computation time of 0.464ms (SD=1.037) for the iteration of a single raw gaze coordinate. This processing time is fast enough for eye trackers with frequencies up to 1200Hz (new raw gaze coordinate each 0.833ms).

5.3 Experiment: Metrics Accuracy

EyeLiveMetrics can only be a valuable eye tracking software tool when its computed eye tracking metrics are accurate and precise. To test that condition, we compare EyeLiveMetrics results to those from Tobii Pro Lab. Therefore, we created a reading experiment ('text stimulus') in Tobii Pro Lab, which is the only stimulus that provides reading metrics. With the internal text editor, we created a text paragraph with the font Arial, a line height of 1.2 and a font size of 48px. AOIs for words are automatically created by the editor. We then recorded a participant reading the text paragraph. After the experiment, raw eye tracking data was exported via the 'data export' functionality. Additionally, we exported fixation- and reading metrics for all words in the stimulus from Tobii Pro Lab as a CSV file via the 'metrics export' functionality. We took a screenshot of the Tobii text stimulus and recreated it as a web page using HTML and CSS. Using a Python script, we load the raw eye-tracking data, replay it and send it to the browser, which displays the reconstructed text stimulus. The EyeLiveMetrics plugin then processes the eye tracking data, maps them to words, and computes metrics. All metrics are stored in a database. After exporting the eye tracking metrics from both software, Tobii Pro Lab and EyeLiveMetrics, we compare them for the following eye tracking metrics:

Fixations: Total Fixation Duration (TFD), Average Fixation Duration (AFD), Minimum Fixation Duration (MiFD), Maximum Fixation Duration (MaFD), Number of Fixations (#F), Timestamp of First Fixation (TFF) and First Fixation Duration (FFD)

Reading: First-Pass First Fixation Duration (FpFFD), First-Pass Duration (FpD), First-pass Regression (FpR), Regression-Path Duration (RPD), Selective Regression-Path Duration (sRPD) and Re-Reading Duration (RRD)

Table 1 shows the comparison results. We report Pearson correlation (ρ) and Mean Absolute Error (MAE) for each fixation- and reading metric. Overall, the fixation metrics have a very high confidence level of similarity between EyeLiveMetrics and Tobii ($0.962 \leq \rho \leq 1.0$) and the MAE on average, has an error margin of < 6 ms. For the reading metrics, we can report similar results ($0.979 \leq \rho \leq 1.0$).

We faced several challenges while comparing fixations on both text stimuli (Tobii Pro Lab & web page): (1) Although using the same font, the same font size, and the same line height, we observed a slightly different rendering of the text in Tobii's internal view and for the web browser. We needed to adapt the line height via CSS to fit Tobii's word bounding boxes. (2) Fixations on spaces between words in Tobii are split into the previous and the next word in Tobii. Experimentally, we found a split of 1/3 of the space to the previous word and 2/3 to the next and implemented that in our code. (3) To have a solid basis for comparison, we have set the fixation filters as simple as possible. Both sources use an I-VT filter with 30 degrees/sec as a threshold. For Tobii, we set the velocity calculator to 3ms so that only the last two coordinates are used for the I-VT calculation. We have switched off all other filters. (4) There are differences in the algorithm between Tobii and EyeLiveMetrics due to the live vs. post-experiment character. For example, EyeLiveMetrics can only evaluate current live coordinates within a queue. This can affect the filter implementation, e.g. for noise filters or the velocity calculator.

Nonetheless, we found a very strong correlation between Tobii's and our fixation and reading metrics, proving that the metrics implementation works accurately and in real time.

6 CONCLUSION

We presented EyeLiveMetrics, an open-source browser plugin to capture gaze data in real time in web environments. Eye coordinates, derived fixations, and saccades are directly mapped to text on webpages. Reading measures are computed live for fixated words, sentences, and texts. This is in contrast to existing eye tracking analysis software that often only allows post-analysis with manual mapping of AOIs and for a limited set of measures. We have shown that the plugin works efficiently for eye trackers with high sampling rates and computes all measures with a high correlation to Tobii Pro Lab measures, as a representative of commercial eye tracking software. Real-time analysis of online reading supports experiments in which participants search, browse, and learn information online. Experimenter have instant access to fixation-, saccade- and reading metrics without the burden of AOI mapping and metrics computation. This additionally allows for live applications that react, for example, to longer fixations or reading difficulties.

ACKNOWLEDGMENTS

This work was partly funded by the DFG, grant no. 525041402; the ECHOES project at GESIS.

REFERENCES

Seoyoung Ahn, Conor Kelton, Aruna Balasubramanian, and Greg Zelinsky. 2020. Towards Predicting Reading Comprehension From Gaze Behavior. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 32, 5 pages. https://doi.org/10.1145/3379156.3391335

David Beymer and Daniel M. Russell. 2005. WebGazeAnalyzer: A System for Capturing and Analyzing Web Reading Behavior Using Eye Gaze. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) *(CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1913–1916. https://doi.org/10.1145/1056808.1057055

Nilavra Bhattacharya, Somnath Rakshit, and Jacek Gwizdka. 2020. Towards Real-Time Webpage Relevance Prediction UsingConvex Hull Based Eye-Tracking Features. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 28, 10 pages. https://doi.org/10.1145/3379157.3391302

Ralf Biedert, Georg Buscher, and Andreas Dengel. 2010a. The eyeBook – Using Eye Tracking to Enhance the Reading Experience. *Informatik-Spektrum* 33, 3 (2010), 272–281.

Ralf Biedert, Georg Buscher, Sven Schwarz, Manuel Möller, Andreas Dengel, and Thomas Lottermann. 2010b. The text 2.0 framework: writing web-based gaze-controlled realtime applications quickly and easily. In *Proceedings of the 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction* (Hong Kong, China) *(EGIHMI '10)*. Association for Computing Machinery, New York, NY, USA, 114–117. https://doi.org/10.1145/2002333.2002351

Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A Robust Realtime Reading-Skimming Classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/2168556.2168575

Efe Bozkir, Gjergji Kasneci, Sonja Utz, and Enkelejda Kasneci. 2022. Regressive Saccadic Eye Movements on Fake News. In *2022 Symposium on Eye Tracking Research and Applications* (Seattle, WA, USA) *(ETRA '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages. https://doi.org/10.1145/3517031.3529619

Georg Buscher, Andreas Dengel, Ralf Biedert, and Ludger V. Elst. 2012. Attentive Documents: Eye Tracking as Implicit Feedback for Information Retrieval and Beyond. *ACM Trans. Interact. Intell. Syst.* 1, 2, Article 9 (jan 2012), 30 pages. https://doi.org/10.1145/2070719.2070722

Carsten Eickhoff, Sebastian Dungs, and Vu Tran. 2015. An Eye-Tracking Study of Query Reformulation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/2766462.2767703

Leticia Bode Emily Vraga and Sonya Troller-Renfree. 2016. Beyond Self-Reports: Using Eye Tracking to Measure Topic and Style Differences in Attention to Social Media Content. *Communication Methods and Measures* 10, 2-3 (2016), 149–164. https://doi.org/10.1080/19312458.2016.1150443

Jacek Gwizdka. 2014. Characterizing Relevance with Eye-Tracking Measures. In *Proceedings of the 5th Information Interaction in Context Symposium* (Regensburg, Germany) *(IIiX '14)*. Association for Computing Machinery, New York, NY, USA, 58–67. https://doi.org/10.1145/2637002.2637011

Daniel Hienert, Dagmar Kern, Matthew Mitsui, Chirag Shah, and Nicholas J. Belkin. 2019. Reading Protocol: Understanding What Has Been Read in Interactive Information Retrieval Tasks. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval* (Glasgow, Scotland UK) *(CHIIR '19)*. Association for Computing Machinery, New York, NY, USA, 73–81. https://doi.org/10.1145/3295750.3298921

Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: from eye fixations to comprehension. *Psychological review* 87, 4 (1980), 329. https://psycnet.apa.org/doi/10.1037/0033-295X.87.4.329

Marcel Adam Just and Patricia A Carpenter. 2018. Using eye fixations to study reading comprehension. In *New methods in reading comprehension research*. Routledge, 151–182. https://doi.org/10.4324/9780429505379-8

Conor Kelton, Zijun Wei, Seoyoung Ahn, Aruna Balasubramanian, Samir R. Das, Dimitris Samaras, and Gregory Zelinsky. 2019. Reading Detection in Real-Time. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 43, 5 pages. https://doi.org/10.1145/3314111.3319916

Diane C. Mézière, Lili Yu, Erik D. Reichle, Titus von der Malsburg, and Genevieve McArthur. 2023. Using Eye-Tracking Measures to Predict Reading Comprehension. *Reading Research Quarterly* 58, 3 (2023), 425–449. https://doi.org/10.1002/rrq.498

Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. 2016. Webgazer: Scalable Webcam Eye Tracking Using User Interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 3839–3845. https://dl.acm.org/doi/10.5555/3061053.3061156

Alex Poole and Linden J Ball. 2006. Eye Tracking in HCI and Usability Research. In *Encyclopedia of human computer interaction*, Claude Ghaoui (Ed.). IGI global, 211–219. https://doi.org/10.4018/978-1-59140-562-7.ch034

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372. https://doi.org/10.1037/0033-2909.124.3.372

Robert W. Reeder, Peter Pirolli, and Stuart K. Card. 2001. WebEyeMapper and WebLogger: Tools for Analyzing Eye Tracking Data Collected in Web-Use Studies. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington) *(CHI EA '01)*. Association for Computing Machinery, New York, NY, USA, 19–20. https://doi.org/10.1145/634067.634082

Erik D Reichle, Alexander Pollatsek, and Keith Rayner. 2012. Using EZ Reader to simulate eye movements in nonreading tasks: A unified framework for understanding the eye–mind link. *Psychological review* 119, 1 (2012), 155. https://psycnet.apa.org/doi/10.1037/a0026473

Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (Palm Beach Gardens, Florida, USA) *(ETRA '00)*. Association for Computing Machinery, New York, NY, USA, 71–78. https://doi.org/10.1145/355017.355028

Siliang Tang, Ronan G Reilly, and Christian Vorstius. 2012. EyeMap: a software system for visualizing and analyzing eye movement data in reading. *Behavior research methods* 44 (2012), 420–438. https://doi.org/10.3758/s13428-011-0156-y

Pablo Valdunciel, Omair Shahzad Bhatti, Michael Barz, and Daniel Sonntag. 2022. Interactive Assessment Tool for Gaze-Based Machine Learning Models in Information Retrieval. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval* (Regensburg, Germany) *(CHIIR '22)*. Association for Computing Machinery, New York, NY, USA, 332–336. https://doi.org/10.1145/3498366.3505834

Michel Wedel and Rik Pieters. 2017. A review of eye-tracking research in marketing. *Review of Marketing Research* 4 (2017), 123–147. https://doi.org/10.1108/S1548-6435(2008)0000004009

Ömer Sümer, Efe Bozkir, Thomas Kübler, Sven Grüner, Sonja Utz, and Enkelejda Kasneci. 2021. FakeNewsPerception: An eye movement dataset on the perceived believability of news stories. *Data in Brief* 35 (2021), 106909. https://doi.org/10.1016/j.dib.2021.106909