EyeLiveMetrics：借助眼动追踪对在线阅读进行实时分析

Daniel Hienert
daniel.hienert@gesis.org GESIS - 莱布尼茨社会科学研究所 德国科隆

Heiko Schmidt
heiko.schmidt@gesis.org GESIS - 莱布尼茨社会科学研究所 德国科隆

Thomas Krämer
thomas.kraemer@gesis.org GESIS - 莱布尼茨社会科学研究所 德国科隆

Dagmar Kern
dagmar.kern@gesis.org GESIS - 莱布尼茨社会科学研究所 德国科隆

arXiv:2601.02044v1 [cs.HC] 5 Jan 2026

摘要

现有眼动追踪软件存在一些局限，尤其是在监测在线阅读方面：(1) 大多数眼动追踪软件将眼动追踪数据记录为原始坐标，将刺激材料记录为屏幕图像/视频，但两者之间没有内在关联。分析人员必须在网页文本上绘制兴趣区（AOI），才能进行更细致的阅读分析。(2) 注视和阅读指标的计算与分析都在实验结束后进行，因此无法用于实时应用程序。

我们介绍 EyeLiveMetrics，这是一款浏览器插件，可实时自动将原始注视坐标映射到文本。该插件可立即计算、存储并提供单词和段落层面的注视、扫视和阅读指标，从而能够即时分析注视行为。我们还讨论了比较评估的结果。EyeLiveMetrics 为测量网络阅读提供了一种灵活的方式，适用于研究实验和实时应用程序。

CCS 概念

• **以人为中心的计算** → **交互系统与工具**；• **信息系统** → **Web应用**。

关键词

在线阅读、阅读指标、实时分析、眼动追踪、浏览器插件

1 引言

眼动追踪是一种成熟的方法，可用于测量和分析不同领域中个体的眼动，例如营销 [Wedel and Pieters 2017],人机交互（HCI）和可用性研究 [Poole and Ball 2006] 以及阅读研究[Just and Carpenter 1980]。这部分是因为眼动能够很好地反映用户的注意力，而且鉴于眼动与认知过程之间的相关性，我们可以据此推断用户的认知状态 [Rayner 1998]。因此，眼动追踪有助于通过实验性用户研究来研究人类行为。然而，大多数眼动追踪系统缺少将原始眼动追踪数据映射到可见刺激的通用功能。通常需要耗时地在每张相关截图上手动绘制兴趣区（AOI），才能完成映射。随着人们越来越关注在网页上将眼动追踪用于阅读相关活动，例如信息检索 [Buscher et al. 2012], 在线新闻消费 [Bhattacharya et al.2020],假新闻检测 [Bozkiret al. 2022]或社交媒体使用[Emily Vraga and Troller-Renfree 2016] 这种工作流程的效率并不高。要在数百个网页上绘制兴趣区，而每个网页又有多个被多位参与者浏览过的兴趣区，需要投入大量时间。

借助 EyeMetricsLive 插件，我们希望为网页刺激材料提供一种解决方案，即浏览器插件，它能够：(1) 实时采集眼动追踪数据；(2) 将这些数据映射到网页上的文本，精确到单词级别；(3) 在记录过程中即时计算眼动追踪指标。该插件使研究人员能够高效分析实验性用户研究中收集的注视数据，也使开发人员能够根据以往的注视行为调整其界面，这与 Bieder 等人提出的“text 2.0 框架”方法类似。 [Biedert 等人, 2010b]例如，网页可以通过自动切换为简易语言来应对阅读理解困难；搜索引擎则可以根据用户迄今阅读的内容推荐搜索词或重新排列搜索结果。此外，该插件会自动将注视数据映射到 HTML 元素，因此无需在眼动追踪实验中耗时手动绘制兴趣区。EyeLiveMetrics 是一款开源模块化插件，只要能够访问眼动追踪数据流，就兼容任何眼动仪。

本文介绍了 EyeLiveMetrics、其流程以及可用指标。此外，我们还探讨了 EyeLiveMetrics 的效度。

2 相关工作

眼动追踪在阅读研究中有着悠久的历史，被认为是可靠的指标，可反映个体随时间变化的认知加工过程 [Just and Carpenter 1980; Rayner 1998]。研究者设计了各种阅读指标来捕捉这些过程，主要依据注视和扫视指标，例如首遍注视时长、首遍回视或重读时长。这些阅读指标的一种常见用途，是判断读者是在深入阅读文本还是在略读 [Bhattacharya et al. 2020; Biedert et al.2012; Gwizdka 2014; Kelton et al. 2019]。尽管多项研究已证明这项分类任务切实可行，但迄今尚未开发出能够实时执行该任务的系统或原型。另一项颇具前景的应用是检测阅读理解情况，例如 [Ahn et al. 2020; Just and Carpenter 2018; Mézière et al. 2023] ； 尤其是在网络假新闻检测领域。相关研究由

Bozkiret al. [Bozkir etal. 2022] 和 Sümeretal. [ÖmerSümer et al. 2021] 的研究表明，阅读包含真实与虚假信息的新闻文章时，会出现不同的眼动模式。

在实时应用程序中，Biedert 等人提出了 Text 2.0 框架 [Biedert 等人，2010b], ，该框架使我们能够实时响应阅读行为。在 Eye Book [Biedert 等人，2010a], 中，用户的注视行为用于判断当前正在阅读哪些文本片段，并据此生成和呈现声音或图像等额外效果。交互式信息检索中的另一种实时方法，则利用注视数据在搜索引擎原型中建议查询改写 [Eickhoff 等人 2015]。在这两个示例中，系统都知道用户当前正在阅读什么，或之前读过什么。

多年来，人们开发了多种软件工具，以支持研究人员开展眼动追踪实验并分析注视数据。一些商业工具（例如 Tobii Pro Lab）支持对实验人员使用集成编辑器创建的特定文本进行阅读指标分析[1] 。随后，这些工具会自动为字符、词语和句子生成兴趣区。然而，对于其他类型的刺激材料（例如网页研究），分析人员仍需手动绘制兴趣区。EyeMap [Tang 等人 2012] 和 Reading-Protocol [Hienert 等人 2019] 是开源工具，它们通过将注视数据映射到基于网页 HTML 元素自动生成的兴趣区，克服了这一缺点。在此基础上，人们开发了一些研究原型，例如 WebgazeAnalyzer [Beymer 和 Russell 2005], 、WebEyeMapper [Reeder 等人 2001] 以及 ReMA [Valdunciel 等人 2022], ，但它们都未考虑实时使用。Webgazer.js [Papoutsaki 等人 2016] 可将眼动数据集成到实时系统中，但它并未考虑底层内容。它根据网络摄像头输入和屏幕上的当前交互行为估算眼动追踪数据，因此会降低准确性。因此，它尚不适用于需要高准确度的场景。EyeLive Metrics 插件正是为此而设计的。它将高精度实时眼动追踪与自动兴趣区检测相结合。

3 EYELIVEMETRICS——浏览器插件

本节将介绍开源浏览器插件 EyeLiveMetrics[2] 的实现。该插件的主要目标是：（1）消除将眼动追踪数据映射到带有兴趣区（AOI）的文本这一耗时过程；（2）即时计算网页文本的注视、扫视或阅读指标；（3）实现对阅读的实时分析，从而在交互场景中能够立即响应检测到的注视行为。下文将概述该插件如何采集眼动追踪数据，以及数据处理流程；该流程包括将注视数据映射到网页文本，并计算眼动追踪指标。

3.1 注视数据采集

EyeLiveMetrics 构建于阅读协议工具 [Hienert 等人 2019], 之上，该工具使用原始眼动数据和已存储的 HTML 页面，在用户实验结束后，通过渲染过程计算对单词的注视。这样一来，坐标便会映射到各个单词的边界框（即兴趣区），随后作为词语眼动注视存储在数据库中。对于使用任意网页开展的网络实验，这一过程容易出错。阅读协议仅在 HTML 页面加载后存储一次，但许多网页都是动态的，会通过 AJAX 调用重新加载内容，或包含菜单、下拉框、轮播组件等动态内容项。如果映射过程使用的网页仅在初次加载后存储一次，就无法反映受试者在整个阅读期间看到的页面状态。因此，眼动追踪坐标映射到兴趣区的过程可能会出现错误。

EyeLiveMetrics 实时执行映射过程。图 1 概述了注视数据的流程。我们编写了一个 Python 脚本，将眼动追踪坐标传输到插件。该脚本通过 Tobii Pro Python SDK[3]访问眼动仪。该 SDK 持续提供带有时间戳的二维显示坐标 (x,y)、三维空间坐标 (x,y,z)，以及用户坐标系中的注视起点和注视位置。Python 脚本还充当 WebSocket 服务器，并将坐标发送到作为客户端的插件。

插件本身实现为 Tampermonkey[4] 用户脚本。Tampermonkey 是一个插件容器，可在任意网页上执行任意 JavaScript 代码。它支持多种浏览器，例如 Chrome 、Edge、Safari、Opera 和 Firefox。它允许基于模式纳入或排除执行用户脚本的网页。

3.2 数据处理流程

图 2 展示了 EyeLiveMetrics 插件中的数据处理流程。WebSocket 客户端接收眼动追踪坐标。该插件使用速度阈值识别（I-VT）算法[Salvucci 和 Goldberg 2000] ，根据眼睛的方向变化速度，将原始注视点分类为注视或扫视。输入参数为注视起点和注视位置坐标，用于计算注视向量并与先前的向量进行比较。算法根据可调节的阈值（例如每秒 30°）返回注视或扫视的分类结果。对于被分类为扫视的注视点，我们会存储其位置和速度。如果分类结果变为注视，则计算最终扫视并将其映射到一个兴趣区。用于扫视映射的兴趣区包括文本段落、图像和视频。如果扫视的起点和终点都位于某个兴趣区内，则会将该扫视保存到该兴趣区中。最后，我们计算扫视时长和长度。我们将在第 3.3 节中详细介绍扫视指标。

分类器识别出一次注视时，便会创建一个注视对象。它包含开始和结束时间、映射到的 HTML 元素（作为兴趣区的单词、图像或视频）、对象边界框以及对象专属指标（例如文本节点中的单词索引/偏移量）等属性。注视结束时，会计算注视和阅读指标。详细概述请参见第 3.3 节。所有数据都存储在浏览器本地，并每五秒或在浏览器标签页状态发生变化时写入数据库，例如关闭或切换标签页时。一般而言，我们会存储参与者 ID、刺激物

[1] https://connect.tobii.com/s/article/How-reading-metrics-work?language=en_US [2] EyeLiveMetrics 是一款采用 GPLv3 许可证的开源软件，网址为 https://git.gesis.org/iir/eyelivemetrics

[3] https://developer.tobiipro.com/python/python-getting-started.html [4] https://www.tampermonkey.net/

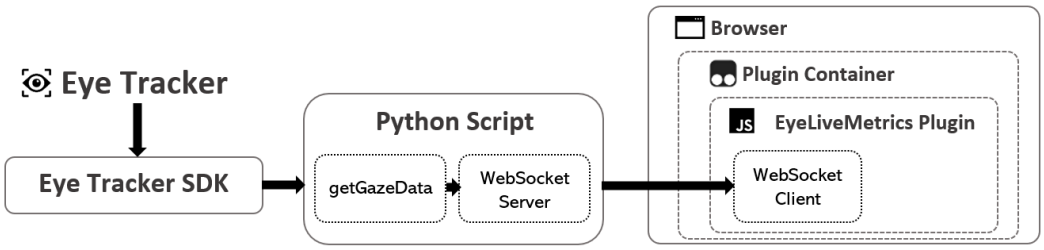


图 1：从眼动仪到 EyeLiveMetrics 插件的注视数据流

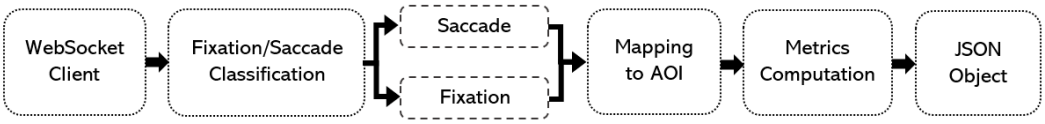


图 2：EyeLiveMetrics 插件中的数据处理

（网页 URL）、开始和结束时间、针对单词、图像和视频的注视与扫视对象，以及整个网页的文本、整个网页文本的词索引和句子索引。

3.3 阅读指标

EyeLiveMetrics 的指标遵循 Tobii Pro Lab[5]对注视、扫视和阅读指标的实现方式。所有可用指标的完整列表可在 EyeLiveMetrics 的 Git 仓库[2] 中找到。

注视是视线在某一点上的持续停留；对于文本刺激而言，视线会停留在词语上，以理解其句法和语义意义 [Reichle 等人 2012]。EyeLiveMetrics 会为每个网页元素（例如词语）存储注视测量数据。首先，我们提供常见的眼动追踪指标，例如总注视时长、注视次数、首次注视的时间戳、首次注视时间、首次注视时长和平均注视时长。此外，我们还提供针对词语的测量数据，例如 HTML 文档中的字符索引和句子编号。最后，我们还提供时间戳、注视组编号和刺激边界框。

扫视是注视之间注视点的快速移动。对于文本刺激，它描述的是从一个单词移动到另一个单词的过程。这意味着，扫视可以抽象地描述文本中的阅读行为。例如，可用于理解段落 [Valdunciel 等人 2022] 的扫描路径， [或用于区分阅读与略读 2012]Biedert 等人。EyeLiveMetrics 会存储文本段落中的扫视数据。对于每次扫视，它都会存储进入和退出扫视的时间戳、顺序索引、时长、长度、幅度、峰值速度以及二维方向向量。

阅读指标可以描述阅读行为的其他特征，例如分析阅读理解 [Mézière 等人 2023]。如果 HTML 元素是一个单词，EyeLifeMetrics 会存储以下阅读指标：首次通读首次注视时长、首遍注视组编号、首遍回视、首次通读时长、回视路径时长、选择性回视路径时长和重读时长。

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

4 EYELIVEMETRICS 的使用

按照项目网站上的说明将 EyeLiveMetrics 插件添加到浏览器，并完成其他必要资源（例如数据库和 Web 服务器）的设置后，即可在用户研究或交互环境中使用该系统。在这两种情况下，都必须使用眼动仪管理软件中通常提供的校准工具，根据用户的眼睛对眼动仪进行校准。接下来可以打开浏览器并激活 EyeLiveMetrics 插件。从此以后，用户每访问一个网页，系统都会进行注视数据采集（见 3.1）。

在用户研究中完成数据收集后，实验人员只需启动分析界面（见图3），即可立即查看分析结果。这也使得数据可以直接用于回顾性访谈。每个单词的注视、扫视和阅读指标都可以在弹出窗口中查看。

在实时应用中，分析后的数据可以立即用于操控界面。例如，图4展示了如何将阅读某个单词的强度直接呈现给读者。在这里，单词的注视时长决定了其背景颜色。尽管这种方法是否有用值得商榷，但它说明了实时生成热图是可行的。

5 评估

我们通过两种不同的方式评估了 EyeLiveMetrics。首先，我们考察了 EyeLiveMetrics 的性能，然后将结果与 Tobii Pro Lab 在相同文本刺激下的结果进行了比较。本节先介绍两项实验所使用的技术环境，随后呈现两项评估。

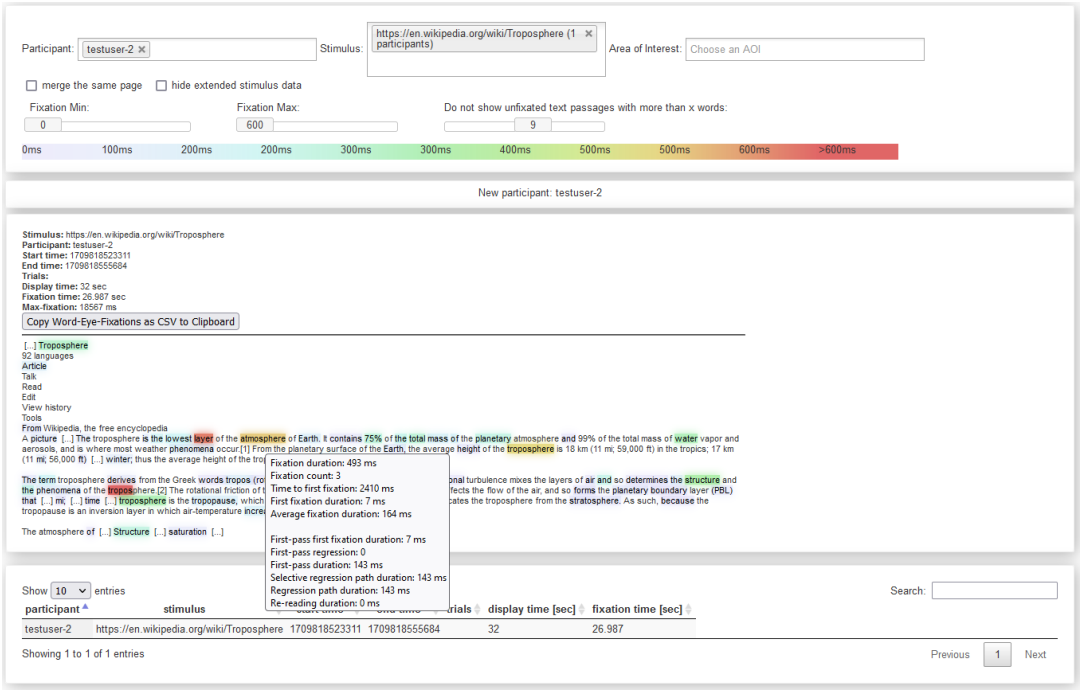


图3：EyeLiveMetric 的用户界面，展示了根据阅读协议用户界面改编的分析结果 [Hienert 等人 2019]。

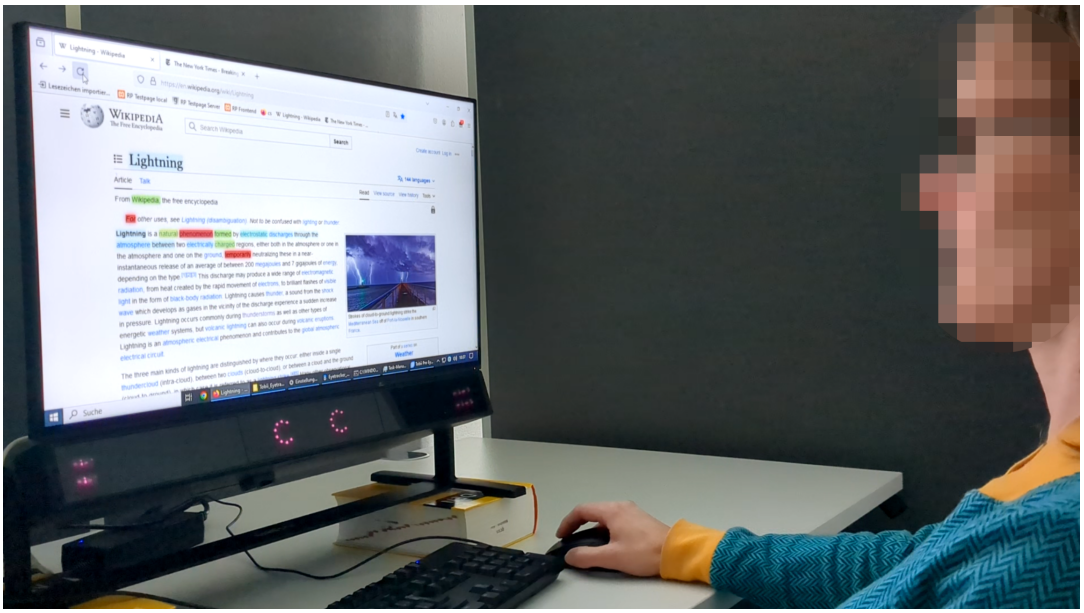


图4：参与者阅读维基百科文章，并实时生成热图。

5.1 技术评估环境

评估中，我们使用了采样率为 300 Hz 的 Tobii Pro Spectrum 设备。采样率限定了 EyeLiveMetrics 这类实时应用程序处理单个注视坐标所能使用的最长时间。两项实验均在一台配备 Intel i7-6700 CPU@3.40GHz 和 16GB 随机存取存储器的 PC 上，使用 Firefox V117.0 作为网页浏览器。如第 3.1 节所述，我们将 EyeLiveMetrics 集成为 Tampermonkey 插件。

表 1：Tobii Pro Lab 与 EyeLiveMetrics 的眼动追踪指标对比。所有时间单位均为毫秒。

指标	注视指标						
	TFD	AFD	MiFD	MaFD	#F	TFF	FFD
皮尔逊	1.0	0.99	0.996	0.962	0.999	1.0	0.973
平均绝对误差 ±标准差	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
	阅读指标						
	FpFFD	FpD	FpR	RPD	sRPD	RRD	
皮尔逊	0.979	1.0	1.0	1.0	1.0	1.0	
平均绝对误差 ±标准差	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 实验：EyeLiveMetrics 性能

由于 EyeLiveMetrics 是一种可实时计算眼动追踪指标的浏览器插件（参见第 3.3 节），其主循环的运行速度至少必须与眼动仪按照其采样率记录并输出原始注视坐标的速度一样快。对于每个坐标，插件都必须接收坐标，将其分类为扫视或注视，然后映射到某个 HTML 元素（文本、单词、图像、视频），最后计算注视、扫视和阅读指标。因此，插件的每次迭代都必须足够快，以处理下一个原始注视数据点。

我们使用第 5.1 节中描述的技术规格进行了性能测试，并记录了在维基百科上阅读一篇长文十分钟的数据。插件记录并处理了 n=176,091 个原始注视坐标，处理单个原始注视坐标迭代的平均计算时间为 0.464 毫秒（标准差为=1.037）。这一处理速度足以满足频率高达 1200 Hz 的眼动追踪器的需求（每 0.833 毫秒产生一个新的原始注视坐标）。

5.3 实验：指标准确性

只有当 EyeLiveMetrics 计算出的眼动追踪指标准确且精确时，它才是有价值的眼动追踪软件工具。为检验这一点，我们将 EyeLiveMetrics 的结果与 Tobii Pro Lab 的结果进行比较。因此，我们在 Tobii Pro Lab 中创建了一个阅读实验（“文本刺激”），因为只有这种刺激物能够提供阅读指标。我们使用内置文本编辑器创建了一个文本段落，字体为 Arial，行高为 1.2，字体大小为 48px。编辑器会自动为词语创建兴趣区。随后，我们记录了一位参与者阅读该文本段落的过程。实验结束后，我们通过“数据导出”功能导出原始眼动数据。此外，我们还通过“指标导出”功能，将 Tobii Pro Lab 中该刺激物所有词语的注视指标和阅读指标导出为 CSV 文件。我们对 Tobii 文本刺激进行了截图，并使用 HTML 和 CSS 将其重建为网页。通过 Python 脚本，我们加载原始眼动追踪数据，重放数据并将其发送到浏览器，由浏览器显示重建后的文本刺激。随后，EyeLiveMetrics 插件会处理眼动追踪数据，将其映射到词语，并计算各项指标。所有指标都存储在数据库中。分别从 Tobii Pro Lab 和 EyeLiveMetrics 导出眼动追踪指标后，我们对以下眼动追踪指标进行比较：

注视：总注视时长（TFD）、平均注视时长（AFD）、最小注视时长（MiFD）、最大注视时长（MaFD）、注视次数（#F）、首次注视时间戳（TFF）和首次注视时长（FFD）

阅读：首次通读首次注视时长 (FpFFD)、首次注视时长 (FpD)、首遍回视 (FpR)、回视路径时长 (RPD)、选择性回视路径时长 (sRPD) 和重读时长 (RRD)

表 1 展示了比较结果。我们报告了每项注视指标和阅读指标的皮尔逊相关系数 (ρ) 以及平均绝对误差 (MAE)。总体而言，EyeLiveMetrics 与 Tobii 的注视指标相似度具有很高的置信水平 ($0.962 \leq \rho \leq 1.0$)，平均平均绝对误差的误差范围为 < 6 毫秒。对于阅读指标，我们也得到了类似的结果 ($0.979 \leq \rho \leq 1.0$)。

比较两种文本刺激材料（Tobii Pro Lab 和网页）上的注视情况时，我们遇到了几项挑战：(1) 尽管使用了相同的字体、字体大小和行高，我们发现 Tobii 内部视图和网页浏览器中的文本渲染略有不同。我们需要通过 CSS 调整行高，使其与 Tobii 的词语边界框相匹配。(2) 在 Tobii 中，落在词语间空格上的注视会被分配到前一个词语和下一个词语。通过实验，我们发现空格的 1/3 会分配给前一个词语，2/3 会分配给下一个词语，并在代码中实现了这一分配方式。(3) 为了确保比较基础可靠，我们尽可能简化了注视过滤器。两个来源均使用 I-VT 滤波器，并将 30 度/秒设为阈值。对于 Tobii，我们将速度计算器设为 3 毫秒，因此 I-VT 计算仅使用最后两个坐标。我们关闭了所有其他滤波器。(4) 由于实时处理与实验后处理的特性不同，Tobii 与 EyeLiveMetrics 使用的算法也有所差异。例如，EyeLiveMetrics 只能在队列中评估当前的实时坐标。这会影响滤波器的实现，例如噪声滤波器或速度计算器。

尽管如此，我们发现 Tobii 的指标与我们的注视和阅读指标之间存在非常强的相关性，证明这些指标的实现既准确又能实时运行。

6 结论

我们介绍了 EyeLiveMetrics，这是一款开源浏览器插件，可在网络环境中实时采集注视数据。眼睛坐标、由此得出的注视和扫视会直接映射到网页上的文本。系统会实时计算参与者注视的词语、句子和文本的阅读指标。这与现有的眼动追踪分析软件形成对比，后者通常只支持事后分析，需要手动映射兴趣区，而且可计算的指标种类有限。我们已经证明，对于采样率较高的眼动仪，该插件能够高效运行，并且其计算出的所有指标都与 Tobii Pro Lab 的指标高度相关；Tobii Pro Lab 可作为商业眼动追踪软件的代表。在线阅读的实时分析支持参与者在线搜索、浏览和学习信息的实验。实验人员无需承担兴趣区映射和指标计算的负担，即可即时获取注视、扫视和阅读指标。此外，该插件还支持实时应用程序，例如对较长的注视或阅读困难作出反应。

致谢

本研究部分由德国科学基金会（资助编号：525041402）和 GESIS 的 ECHOES 项目资助。

参考文献

Seoyoung Ahn、Conor Kelton、Aruna Balasubramanian 和 Greg Zelinsky。2020年。《基于注视行为预测阅读理解》。载于 *ACM*眼动追踪研究与应用研讨会（德国斯图加特）（*ETRA '20* 短文）。计算机协会，美国纽约州纽约市，文章编号32，共5页。https://doi.org/ . / . David Beymer 和 Daniel M. Russell。2005年。《WebGazeAnalyzer：一种利用眼动捕捉并分析网页阅读行为的系统》。载于 *CHI '05*人机交互系统中的人为因素扩展摘要（美国俄勒冈州波特兰）（*CHI EA '05*）。计算机协会，美国纽约州纽约市，1913–1916。https://doi.org/10.1145/1056808.1057055 Nilavra Bhattacharya、Somnath Rakshit 和 Jacek Gwizdka。2020年。《利用基于凸包的眼动追踪特征实现实时网页相关性预测》。载于*ACM*眼动追踪研究与应用研讨会（德国斯图加特）（*ETRA '20* 附属会议）。计算机协会，美国纽约州纽约市，文章编号28，共10页。https://doi.org/10.1145/3379157.3391302 Ralf Biedert、Georg Buscher 和 Andreas Dengel。2010a。《the eyeBook——利用眼动追踪提升阅读体验》。《*Informatik-Spektrum*》33卷3期（2010年），272–281。Ralf Biedert、Georg Buscher、Sven Schwarz、Manuel Möller、Andreas Dengel 和 Thomas Lottermann。2010b。《text 2.0 框架：快速、轻松地编写基于网页的注视控制实时应用程序》。载于*2010*年智能人机交互眼动研讨会论文集（中国香港）（*EGIHMI '10*）。计算机协会，美国纽约州纽约市，114–117。https://doi.org/10.1145/2002333.2002351 Ralf Biedert、Jörn Hees、Andreas Dengel 和 Georg Buscher。2012年。《稳健的实时阅读-略读分类器》。载于眼动追踪研究与应用研讨会论文集（加利福尼亚州圣巴巴拉）（*ETRA '12*）。计算机协会，美国纽约州纽约市，123–130。https://doi.org/10.1145/ 2168556.2168575 Efe Bozkir、Gjergji Kasneci、Sonja Utz 和 Enkelejda Kasneci。2022年。《假新闻中的回归性扫视眼动》。载于*2022*年眼动追踪研究与应用研讨会（美国华盛顿州西雅图）（*ETRA '22*）。计算机协会，美国纽约州纽约市，文章编号7，共7页。https://doi.org/10.1145/3517031.3529619 Georg Buscher、Andreas Dengel、Ralf Biedert 和 Ludger V. Elst。2012年。《注意性文档：作为信息检索及其他领域隐式反馈的眼动追踪》。《*ACM*交互式智能系统汇刊》1卷2期，文章编号9（2012年1月），共30页。https://doi.org/10.1145/2070719.2070722 Carsten Eickhoff、Sebastian Dungs 和 Vu Tran。2015年。《查询改写的眼动追踪研究》。载于第*38*届*ACM*国际信息检索研究与开发会议论文集（智利圣地亚哥）（*SIGIR '15*）。计算机协会，美国纽约州纽约市，13–22。https: // doi.org/10.1145/2766462.2767703 Leticia Bode、Emily Vraga 和 Sonya Troller-Renfree。2016年。《超越自我报告：利用眼动追踪衡量社交媒体内容的主题与风格差异对注意力的影响》。《传播方法与测量》10卷2-3期（2016年），149–164。https://doi.org/10.1080/19312458.2016.1150443 Jacek Gwizdka。2014年。《使用眼动追踪指标刻画相关性》。载于第五届情境信息交互研讨会论文集（德国雷根斯堡）（*IIiX '14*）。计算机协会，美国纽约州纽约市，58–67。https://doi.org/10.1145/2637002.2637011 Daniel Hienert、Dagmar Kern、Matthew Mitsui、Chirag Shah 和 Nicholas J. Belkin。2019年。《阅读协议：了解交互式信息检索任务中已阅读的内容》。载于*2019*年人类信息交互与检索会议论文集（英国苏格兰格拉斯哥）（*CHIIR '19*）。计算机协会，美国纽约州纽约市，73–81。https://doi.org/10.1145/3295750. 3298921 Marcel A Just 和 Patricia A Carpenter。1980年。《阅读理论：从眼注视到理解》。《心理学评论》87卷4期（1980年），329。https://psycnet.apa.org/ doi/10.1037/0033-295X.87.4.329 Marcel Adam Just 和 Patricia A Carpenter。2018年。《利用眼注视研究阅读理解》。载于《阅读理解研究新方法》。劳特里奇，151–182。https://doi.org/10.4324/9780429505379-8 Conor Kelton、Zijun Wei、Seoyoung Ahn、Aruna Balasubramanian、Samir R. Das、Dimitris Samaras 和 Gregory Zelinsky。2019年。《实时阅读检测》。载于第十一届*ACM*眼动追踪研究与应用研讨会论文集（科罗拉多州丹佛）（*ETRA '19*）。计算机协会，美国纽约州纽约市，文章编号43，共5页。https://doi.org/10.1145/3314111.3319916 Diane C. Mézière、Lili Yu、Erik D. Reichle、Titus von der Malsburg 和 Genevieve McArthur。2023年。《使用眼动追踪指标预测阅读理解》。《阅读研究季刊》58卷3期（2023年），425–449。https://doi.org/10.1002/rrq.498 Alexandra Papoutsaki、Patsorn Sangkloy、James Laskey、Nediyana Daskalova、Jeff Huang 和 James Hays。2016年。《Webgazer：利用用户交互实现可扩展的网络摄像头眼动追踪》。载于第二十五届国际人工智能联合会议论文集（美国纽约州纽约市）（*IJCAI'16*）。AAAI出版社，3839–3845。https://dl.acm.org/doi/10.5555/3061053.3061156 Alex Poole 和 Linden J Ball。2006年。《人机交互与可用性研究中的眼动追踪》。载于《人机交互百科全书》，Claude Ghaoui（编）。IGI Global，211–219。https://doi.org/10.4018/978-1-59140-562-7.ch034 Keith Rayner。1998年。《阅读与信息处理中的眼动：20年研究》。《心理学通报》124卷3期（1998年），372。https://doi.org/10.1037/0033- 2909.124.3.372 Robert W. Reeder、Peter Pirolli 和 Stuart K. Card。2001年。《WebEyeMapper和WebLogger：分析网络使用研究中收集的眼动追踪数据的工具》。载于*CHI '01*人机交互系统中的人为因素扩展摘要（华盛顿州西雅图）（*CHI EA '01*）。计算机协会，美国纽约州纽约市，19–20。https://doi.org/10.1145/634067.634082 Erik D Reichle、Alexander Pollatsek 和 Keith Rayner。2012年。《使用EZ Reader模拟非阅读任务中的眼动：理解眼—心联系的统一框架》。《心理学评论》119卷1期（2012年），155。https: // psycnet.apa.org/doi/10.1037/a0026473 Dario D. Salvucci 和 Joseph H. Goldberg。2000年。《在眼动追踪协议中识别注视和扫视》。载于*2000*年眼动追踪研究与应用研讨会论文集（美国佛罗里达州棕榈滩花园）（*ETRA '00*）。计算机协会，美国纽约州纽约市，71–78。https://doi.org/10.1145/ 355017.355028 Siliang Tang、Ronan G Reilly 和 Christian Vorstius。2012年。《EyeMap：用于可视化和分析阅读中眼动数据的软件系统》。《行为研究方法》44卷（2012年），420–438。https://doi.org/10.3758/s13428-011-0156-y Pablo Valdunciel、Omair Shahzad Bhatti、Michael Barz 和 Daniel Sonntag。2022年。《信息检索中基于注视的机器学习模型交互式评估工具》。载于*2022*年人类信息交互与检索会议论文集（德国雷根斯堡）（*CHIIR '22*）。计算机协会，美国纽约州纽约市，332–336。https://doi.org/10.1145/3498366.3505834 Michel Wedel 和 Rik Pieters。2017年。《营销领域眼动追踪研究综述》。《营销研究综述》4卷（2017年），123–147。https://doi.org/10.1108/S1548- 6435(2008)0000004009 Ömer Sümer、Efe Bozkir、Thomas Kübler、Sven Grüner、Sonja Utz 和 Enkelejda Kasneci。2021年。《FakeNewsPerception：关于新闻报道感知可信度的眼动数据集》。《数据简报》35卷（2021年），106909。https://doi.org/10. 1016/j.dib.2021.106909