Splitting the Web Analytics Atom: From Page Metrics and KPIs to Sub-Page Metrics and KPIs

Ilan Kirsh
kirsh@mta.ac.il
The Academic College of Tel Aviv-Yaffo
Tel-Aviv, Israel

Mike Joy
M.S.Joy@warwick.ac.uk
University of Warwick
Coventry, United Kingdom

ABSTRACT

Web analytics Key Performance Indicators (KPIs) are important metrics used to evaluate websites and web pages against objectives. The power of KPIs is in their simplicity. Every web page can be assessed by numeric KPI values, which can be easily calculated, compared, and tracked over time. KPIs highlight the strengths and weaknesses of individual web pages and significantly help in maintaining, improving, and optimizing websites. Current web analytics metrics and KPIs, in academic studies as well as in commercial tools, relate to entire websites and web pages. This paper advocates extending KPIs use to sub-page elements, such as paragraphs, as an effective way to refine knowledge and leverage web analytics capabilities. We discuss the potential and challenges of sub-page web analytics and define a framework for calculating sub-page metrics from accumulated in-page user activity data, such as mouse and keyboard events. Then we propose potential KPIs that may be effective in highlighting the strengths and weaknesses of individual page parts, such as paragraphs. We use web usage data from a sample website to demonstrate these ideas. This study is the first step towards sub-page web analytics metrics and KPIs. Further work is required in order to gain more knowledge about potential KPIs that are introduced in this work, as well as to explore new methods, metrics, and KPIs.

CCS CONCEPTS

• **Information systems → Web mining**; **Traffic analysis**; *Browsers*.

KEYWORDS

Web Analytics, Web Usage Mining, Data Mining, Web Activity, Analytics Tools, Websites, Web Pages, Web Content, Web Users, Paragraphs, Online Learning, Educational Technology, Javascript, AJAX, Browsers

ACM Reference Format:
Ilan Kirsh and Mike Joy. 2020. Splitting the Web Analytics Atom: From Page Metrics and KPIs to Sub-Page Metrics and KPIs. In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS'20). ACM, Biarritz, France.* https://doi.org/10.1145/3405962.3405984

1 INTRODUCTION

Maintaining, improving, and optimizing a website require a good understanding of how the website is used by its visitors. Web analytics tools collect, analyze, and present web usage data, in ways that can highlight what in the website works well, and possibly can be leveraged further, and what might not work so well, and may require fixing, improvement, or removal. Raw usage data, such as web server access log files and click-stream logs, may contain too much information to be useful unprocessed for most purposes. Web analytics tools use metrics to summarize web usage data as simple numbers that are easy to work with. In this context, a web analytics Key Performance Indicator (KPI) [18, 19] is a valuable metric that helps to verify whether the objectives of a website are achieved.

Objectives vary from one website to another, from one page to another on the same website, and sometimes even from one paragraph to another on the same web page, so KPIs are context-dependent. Table 1 shows possible basic KPIs.

Table 1: Sample Web Analytics Metrics / KPIs

Scope	Positive Indicators	Negative Indicators
Website	Returning Visitors Rate	Bounce Rate
Page	Page views, Active Time	Exit Rate

On the website level, a high rate of returning visitors is usually considered a positive indication that the website is useful and attractive. A high bounce rate, which is the percentage of visitors that leave the website quickly after they arrive without activity, is a negative signal.

On the page level, a high number of page views indicates that the page is popular. A long average activity (or engagement) time of users on a specific page (per page view) may indicate that the page is interesting and useful. On the other hand, a high exit rate, which is the percentage of views of the page that are immediately followed by leaving the website, may indicate that there is something wrong with the page.

Web analytics metrics and KPIs are defined for websites and web pages. To the best of our knowledge, no prior study explored KPIs for sub-page elements (except estimated attention time, as discussed in section 2 below). In this study we:

- discuss possible reasons why web pages are treated as indivisible atoms, and how splitting pages to sub-elements and using web analytics to evaluate page parts, such as sections and paragraphs, could be beneficial;
- describe and formalize a model for sub-page web analytics, and propose and discuss potential metrics and KPIs;

- demonstrate sub-page metrics and KPIs using real web usage data from a sample website.

Web analytics is one of the most widely used applications in the field of web usage mining. It is essential for everyone that is involved in website management and maintenance, including web developers, web designers, web administrators, and content editors. In this study, we concentrate on web analytics of website content, and accordingly, we focus on metrics and KPIs that highlight which content on the website is satisfactory and useful, and which content may require improvement or removal. We use real usage data from a live technical-educational website for evaluation and demonstration.

2 RELATED WORK

Web analytics methods and tools were found to be very effective in understanding how visitors use websites and in optimizing and improving websites and digital services. This has been demonstrated in a wide range of areas and applications, including, for example, in e-commerce [12], digital marketing [3, 17], online news [16], and online learning [28]. As discussed in the introduction above, various metrics and KPIs are used in web analytics to summarize web usage data in manageable and productive ways. These metrics and KPIs are defined, explained, and evaluated in several web analytics books [1, 7, 18, 19].

2.1 In-Page Engagement

Various methods have been suggested to estimate the interest of a user in a web page. The elapsed time (or dwell time), in which a page is open in the browser, is considered as a weak indicator of user interest. Activity or engagement time, which is the time in which the user is actively engaged with the page, is considered as a better indicator. An early work by Claypool et al. [4] found that the amount of scrolling on a page has a strong correlation with explicit user interest. In those early years of the web, researchers had to develop and use special browsers in order to be able to capture user scrolling and mouse actions. Today's modern browsers expose such user actions through JavaScript as a standard, so this barrier was removed. Guo and Agichtein [11] found that mouse movements and scrolling actions are significantly more effective than page dwell time in estimating the relevance of search results to users. The viewport (or scrolling) position of the page in the browser window reflects which parts of the document are visible to the user at each point in time. Lagun and Lalmas [23] suggested four metrics for online news article pages, based on viewport data. Smadja et al. [32] used viewport data to identify backtracking patterns (scrolling up or backward on the page) in reading online news articles. Conlen et al. [5] developed visualization solutions for presenting reading patterns of interactive articles on the web, based on viewport position and scrolling. The focus of all these studies was on finding better metrics, KPIs, or visualization methods, for understanding and evaluating web pages, rather than treating page parts as standalone units of interest.

2.2 User Attention Position

To be able to analyze parts of pages individually, we need a way to know where the user's attention is focused. Eye tracking is often used to analyze user attention during various activities, including for example, in reading documents [2, 33], performing tasks [15], interacting with ads [26], and using mobile devices [27]. Usually, we do not have eye tracking and gaze data available for web analytics, so we need alternative ways to obtain user attention information.

We cannot estimate the user's attention at every given moment without eye tracking data. Therefore, Grusky et al. [9] suggested attention evaluation functions that assign user attention probability to page parts at each point in time. Comparing these functions with knowledge on average reading speeds in different languages, they found out that a function that divides the user attention between all the visible page parts in the viewport, using the normal distribution (i.e. the parts in the middle of the viewport receive larger shares of attention), is the most accurate. Several studies found a connection between eye gaze positions (and user attention in general) and certain user actions, mainly mouse events [4, 10, 14, 20, 29–31]. Based on this principle, Hauger et al. [13] developed a function that evaluates the total reading time of each part of the page. Tuning of that function (i.e. setting weights to different indicators) was done by matching to eye tracking data.

In-page user activity and attention can be aggregated and visualized using heatmaps [21, 24, 25]. This is supported by premium commercial web analytics tools, such as Crazy Egg, MouseFlow, and Hotjar [21].

Given that various methods to estimate and visualize user attention on parts of web pages have already been suggested, a natural next step should be to explore more advanced sub-page metrics and KPIs, beyond user attention and reading time metrics. Such KPIs may tell us more about how parts of a web page function relative to the objectives. To the best of our knowledge, no such work has been done or published yet.

3 TOWARDS SUB-PAGE WEB ANALYTICS

In this section, we discuss why web analytics metrics and KPIs are not available for page parts such as paragraphs, and how extending web analytics to sub-page elements could be beneficial.

3.1 Why Sub-Page Web Analytics is Not Used

We can think of several possible reasons why pages are treated as undivided atoms in web analytics, as follows:

- Historically, in the early days of the World Wide Web, statistics and web analytics tools were based on server-side data, mainly from the web server logs, in which requests for pages are the only available data. This limitation was removed by modern browsers, but web analytics has not been fully upgraded yet to take advantage of all the new abilities.
- Although modern browsers expose (via JavaScript) many in-page user actions, the most important information, which is what exactly is read by the user at each time, is unavailable. Therefore, attributing most user actions to specific sub-page elements may be challenging and less accurate than attributing actions to web pages. We discuss this challenge later in this paper.
- Conventional page-level web analytics is already a big data application. Extending data collection and analysis further to micro in-page user actions would increase the amount of

collected and processed data significantly, and accordingly, would also increase costs.

- For most purposes, page-level web analytics is effective, and it generally provides sufficient value.

The last two points are especially important. If page-level web analytics is usually effective, and it requires fewer resources than sub-page web analytics, then maybe there is no need for sub-page web analytics at all? In the next section, we try to make the case for sub-page web analytics.

3.2 The Benefits of Sub-Page Web Analytics

Although sub-page web analytics requires more resources, it does not mean that it is impractical. The rapid decline in data storage costs over time and the increase in computer power and efficiency make sub-page web analytics reachable and affordable. The move to cloud computing in recent years eliminates technical barriers in the processing of big data. As a matter of fact, there are already commercial web analytics services that record and present low-level, sub-page usage data. This is not the mainstream yet, and free web analytics services such as Google Analytics (which leads the market [6]) currently do not support it. However, there are many commercial web analytics services that record mouse clicks and movements and use them to show user attention on areas of pages [21]. In these premium services, the data are available and used for visualization (e.g. for attention heatmaps, as discussed in section 2). These data could be used for sub-page web analytics metrics and KPIs as well.

We believe that sub-page web analytics could be very beneficial. Obviously, website-level metrics and KPIs on their own (with no page-level metrics) are insufficient. They can provide very general estimates on the website as a whole, but in order to improve and optimize a website, we need the finer details that page-level metrics and KPIs provide. Otherwise, we cannot tell which pages are more successful and which pages require improvement. The same logic should also apply to sub-page metrics and KPIs. The knowledge that a page requires improvement is too general. That page may have good and bad parts, and without sub-page metrics and KPIs, it is very difficult to know which parts of the page function properly and which parts require adjustments. Therefore, although page-level web analytics is effective, sub-page metrics and KPIs may make web analytics even more effective.

Sub-page web analytics can be useful for most types of websites, as web analytics in general is useful for most types of websites. More focused information about page parts can always provide a better picture. It is like increasing the resolution of a satellite image, exposing things that are invisible in a lower resolution. In some fields, it may be more beneficial than in others. For example, on websites that include long-term content, such as online learning materials, investment in content improvement is essential so sub-page web analytics can help. Using sub-page web analytics we may be able to locate an unclear paragraph that requires rephrasing or content that visitors find unattractive, based on metrics that reflect collective user reactions. Similarly, any other website that contains pages with relatively constant content and consistent long-term interest can benefit from sub-page web analytics. This includes websites with rules and regulations, technical information, health information, etc. On the other hand, on websites that contain mainly short-term content, such as online news websites, using sub-page web analytics to improve content may not be economical, as the content on the front pages is updated regularly. Nonetheless, sub-page web analytics can also be beneficial to some extent on these websites, for example, in analyzing fixed page elements that every page on the website contains (e.g. headers, menus, footers, etc.).

4 SUB-PAGE WEB ANALYTICS MODEL

In this section, we set the foundations of sub-page web analytics. We define, explain, and formalize basic concepts in the context of sub-page web analytics, including page elements and parts, page views, attention functions, metrics, objectives, goals, and KPIs.

4.1 Sub-Page Parts

Most web pages are written in HTML. An HTML page has a natural tree-like structure of elements. Text content is embedded mainly in header elements (H1, H2, etc.), paragraph elements (P), list elements (UL, OL, LI), and table elements (TABLE, TR, TH, TD, etc.). Code is embedded mainly in preformatted elements (PRE). Content is embedded between the element's opening and closing tags. Breaking an HTML page into elements is easy using HTML parsers, which are available for all the popular programming languages.

This study focuses on page content, so fixed page parts that repeat on every page, such as the page header, footer, and menus, were not analyzed. These page parts, however, may be of interest in other studies. In sub-page web analytics we are analyzing page parts. A page part can be an HTML element, e.g. a paragraph, or a group of HTML elements, such as a section that contains several paragraphs.

We mark the set of all the relevant page parts that we use in sub-page web analytics as $P = \{p_1, \ldots, p_k\}$.

4.2 Page Views and Events

Web usage data are organized as page views $V = \{v_1, v_2, \ldots, v_n\}$. Each page view, v, represents a single view of a page by a user, and is associated with a list of events, $events(v) = \{e_1 = (t_1, a_1, p_1), \ldots, e_m = (t_m, a_m, p_m)\}$, representing user activity that was tracked and collected (mainly using client-side JavaScript) during the time that the page was open in the user's browser. Every event $e = (t, a, i)$ contains a timestamp, t, an event type, a (e.g. MouseMove), and additional information, i, on the event (e.g. cursor positions for mouse events). The last event in each page view's event list is usually of type UNLOAD, indicating leaving the page. The timestamp t is between 0, which represents the time when the page was loaded, and $duration(v)$, which indicates the time when it was unloaded. In conventional web analytics, sequences of page views by the same user are grouped into visits or sessions, but such grouping is not needed in this study.

Modern browsers expose, using client-side JavaScript, details about many user action events. Relevant data can be sent by the website client-side code to the back-end server (through AJAX) for collection, processing, and analysis. Table 2 lists a small subset of these events.

Table 2: Sample Client-Side Events

Event	Description
Home	Key pressed for moving to the page top.
ScrollUp	Viewport scrolled up (backward).
ScrollDown	Viewport scrolled down (forward).
MouseMove	Mouse moved.
MouseOver	Mouse cursor moved over a link.
MouseRight	Mouse moved to the right.
MouseLeft	Mouse moved to the left.
Click	Mouse button clicked.
DblClick	Mouse button double clicked.
Copy	Selected text copied to the clipboard.
Ctrl+F	Key pressed for in-page search (Windows).
Meta+F	Key pressed for in-page search (Mac).
Search	Search box used to search the website.
Blur	Page loses the keyboard focus.
MouseLeave	Mouse cursor leaves the page.
Hide	Page becomes hidden (browser tab is switched).
Unload	Page unloaded (browser or tab is closed).

Most of the events above are simple JavaScript events (e.g. MouseMove, Blur, Unload). Some are based on filtering JavaScript events by checking parameters or previous state (e.g. MouseRight).

In conventional page-level web analytics, each page view is associated with a single web page. It is more complicated in sub-page web analytics because each page view is associated with multiple page parts. Some events (such as mouse clicks) are linked to specific page parts, but most others (for example, pressing keys and scrolling) are not. Therefore, we need a way to associate events, and user attention in general, with specific page parts. This challenge of locating user attention without eye tracking data was already mentioned above and we discuss it now in more detail.

4.3 User attention Functions

We need an attention function of the form:

$$attention(v, t, p) \tag{1}$$

that evaluates the probability that at time t in page view v, the attention of the user was on page part p. The *attention* function should return a value in the range of $[0, 1]$. It should be 0 for page parts that are outside of the viewport at time t, and also 0 for any t value at which the page is inactive or already unloaded.

The *attention* function should return 1 (representing 100%) to a page part if there is high certainty that the user's attention was on that page part, or otherwise, split the 1.0 value between several page parts that are visible in the viewport at time t. In subsection 2.2, we mentioned two main approaches for estimating user attention:

(1) Dividing attention probability among the page parts that are visible in the viewport [9];
(2) Estimating the attention position by various events, including mouse activity [13].

Viewport-only attention functions are less accurate as it is impossible to know (without using additional indicators) which of the visible parts in the viewport captures the user's attention. They are mainly useful when no mouse position data are available. The following attention position indicators were used in this study:

(1) Users select text (including code);
(2) Users click the mouse (on a link or to start selection);
(3) Users move the mouse cursor over links;
(4) Users move the mouse elsewhere.

Based on these indicators, three attention functions have been examined: ***attention3***, ***attention4***, and ***attention5***. *attention3* uses the first three indicators. *attention4* uses all four indicators. Indicator (4), which is moving the mouse with no other action, is considered less accurate in indicating the focus of the user's attention. Both *attention3*, and *attention4* calculate the attention of part p in page view v at time t, by the following steps:

(1) Find the nearest attention position indicator event (in time, before or after t), its time t_i and its page part p_i;
(2) If $p \neq p_i$ return 0;
(3) If $|t - t_i| > 5$ seconds return 0;
(4) Otherwise return 1.

Functions *attention3* and *attention4* return either 0 or 1. Only a single paragraph can get 1 at any point in time. We assume that an attention position indicator is likely to imply position for at least a short time frame, and five seconds before and after the event seems to work reasonably. These attention functions are quite different from existing attention functions by returning 0 if there is no attention position indicator nearby. In other words, when there is no relatively high confidence about the user's attention then these functions do not guess.

Users can read many paragraphs in a sequence without generating any of the four attention position indicators, so using these attention functions may not provide attention position most of the time. Therefore, an additional function, *attention5*, was examined. It is based on *attention4* but adds a fallback. Any time that is not covered by indicators is divided and allocated as attention time to the visible parts in the viewport, using the normal distribution, following [9].

The different attention functions demonstrate a trade-off between quantity (more attention time is allocated) and quality (the allocation is more accurate).

4.4 Sub-Page Metrics

The first metric that we can now define is the total attention time of a page part, p, in all the inspected page views:

$$attentionTime(p) = \sum_{v \in V} \int_{t=0}^{duration(v)} attention(v, t, p)\, dt \tag{2}$$

Note, that it is not necessarily an evaluation of the actual attention time. It may be the total time for which we have high certainty that the user's attention is on part p.

We can also use the *attention* function to count occurrences of a specific event type, a, on a page part p, in all the page views:

$$eventCount(a, p) = \sum_{v \in V} \sum_{(t,a,i) \in events(v)} attention(v, t, p) \tag{3}$$

Event counting requires iteration over all the page views (the external $\sum$), and for each page view iteration over all its events of type a (the internal $\sum$). Because we are interested only in events that are associated with a specific page part, p, counting an event depends on the user's attention on page part p at time t, which is a value between 0 and 1.

Normally, we are less interested in absolute numbers, but rather in the frequency of events on a page part, relative to attention time:

$$eventFreq(a,p) = \frac{eventCount(a,p)}{attentionTime(p)} \tag{4}$$

All the client-side JavaScript events in Table 2 can be used in frequency metrics of this type. We use event names as shortcuts to the frequency metrics, so for example, the $eventFreq(MouseMove, p)$ metric can be simply referred to as MouseMove.

Another form of metric that could be useful is the ratio between counts (and frequencies) of two event types on a page part:

$$eventRatio(a_1, a_2, p) = \frac{eventCount(a_1, p)}{eventCount(a_2, p)} \tag{5}$$

Two Sample ratio metrics of type (5) are shown in Table 3:

Table 3: Sample Event Ratio Metrics

Name	Ratio
Scroll-DU	$eventRatio(ScrollDown, ScrollUp, p)$
Mouse-RL	$eventRatio(MouseRight, MouseLeft, p)$

The Scroll-DU metric measures the scrolling down (forward) to scrolling up (backward) event count ratio. Regularly, we expect to have more ScrollDown events than ScrollUp, as scrolling down reflects ordinary reading order.

Similarly, the Mouse-RL metric measures the ratio between the time that the mouse moves to the right, and the time that the mouse moves to the left (mouse events are sampled at a constant rate, so in practice, it is calculated as the ratio of event counts).

4.5 Objectives, Goals, and KPIs

Objectives are context-dependent. A primary objective of an e-commerce website may be to maximize sales. A primary objective of an online news website may be to maximize user clicks on ads. For an educational-informative website, such as Wikipedia, the primary objective could be to provide high quality and useful information for everyone. Objectives can be global for a website, local at the level of pages, or even at the sub-page level. A local objective of paragraphs in Wikipedia may be to provide useful and easy to read information, which fits the page content and integrates contextually with neighboring paragraphs.

Objectives that can be measured automatically, such as online sales and clicks on ads are referred to in web-analytics terminology as goals [18, 19]. Objectives are not always easy to measure. As discussed in subsection 3.2, online learning websites are a main target for sub-page web analytics. In these websites, a primary objective is to provide high quality and useful information. There are no clear measurable goals such as sales or ad clicks in this context. User feedback can help in spotting content quality issues, but it is often missing or incomplete, and the main purpose of web-analytics is to provide perspectives based on statistical data, which can be collected automatically with minimal costs.

KPIs are metrics that help in evaluating websites and web pages against objectives. In websites with no clear measurable goals, the role of KPIs is even more important, as good KPIs may compensate for the absence of measurable goals, by providing alternative ways to measure success automatically. Similar to KPIs at the website level, sub-page KPIs are valuable sub-page metrics, which can indicate success or failure in achieving objectives.

Any metric of type (4) with an event from Table 2 is a potential KPI. For example, a high frequency of Hide and Unload events may be a negative indicator, similar to a high exit rate at the page level. It may signal issues with a specific paragraph that causes too many users to leave. But as with the Exit Rate KPI, this is not decisive. Leaving the page could also indicate that users found what they were looking for (e.g. answers to their questions on websites such as StackOverflow). Therefore, each metric has to be examined and evaluated in the context of particular websites.

KPIs may also be metrics of type (5). For example, a high value of Mouse-RL may indicate frequent reading activity [20], which is a good signal that may indicate user interest in the page content [22]. A low value of Scroll-DU may indicate intensive backtracking (scrolling up), which is known as a bad signal at the page level [32]. Using these values as sub-page metrics for paragraphs (rather than as page-level values) could be useful in sub-page web analytics. Again, all these metrics require further investigation.

5 EXPERIMENTAL RESULTS

This section presents experiments on calculating sub-page metrics and KPIs for page parts of a real website. A full evaluation of KPIs is out of the scope of this work and probably justifies intensive dedicated research work for each KPI separately, across different websites and conditions. The goal of these experiments is more modest: a preliminary evaluation of the feasibility of the proposed model, and a demonstration of the potential of sub-page web analytics with a few examples.

5.1 Implementation and Usage Data

As discussed in section 3, sub-page web analytics may be especially effective in evaluating long-term content, such as online learning materials. We examined sub-page web analytics metrics and KPIs on the manual pages of ObjectDB (www.objectdb.com), an object-oriented database system. ObjectDB is based on the Java Persistence API (JPA), the standard API in Java for accessing databases in an object-oriented way, and its manual is a popular source of information for JPA users, for learning and as a reference.

To collect web usage data, all the pages of the ObjectDB manual were linked to special client-side JavaScript tracking code. When a web page was loaded into the user's browser, the JavaScript code ran on the browser in the background, captured relevant events, and reported back to the server. Collected data were stored anonymized in a dedicated database on the server (adhering to industry standards of data anonymization and user privacy preservation) and were used in the sub-page web analytics experiments. Figure 1 shows the general architecture of this implementation.

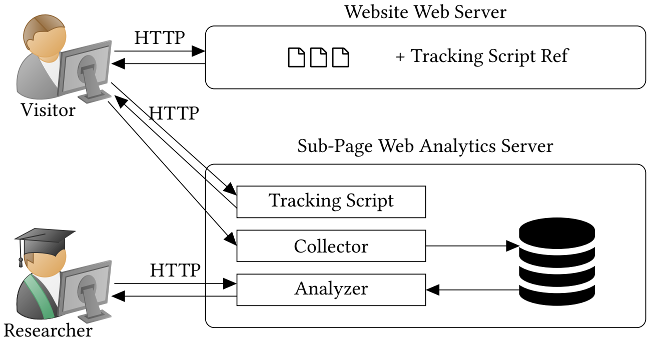

Figure 1: Sub-Page Web Analytics Implementation

The ObjectDB manual contains 69 web pages. Figure 2 shows the top of one of these pages.

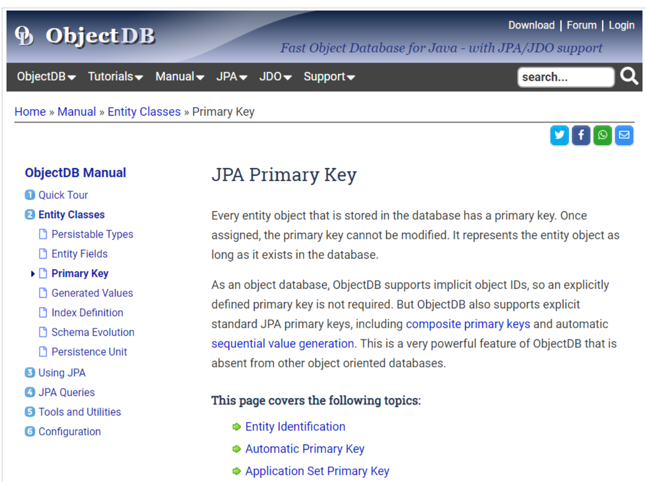

Figure 2: A Manual Page on the ObjectDB Website

The content of each web page was divided into parts by parsing the page and extracting key HTML elements (P, PRE, etc.), as described in subsection 4.1. The 69 pages contain 1,972 parts of different types, as shown in Table 4. The experiments focused on the 1,149 text paragraphs (in P elements) and the 261 fragments of Java and JPQL (Java Persistence Query Language) code (in PRE elements).

Table 4: General Details on the 69 Web Pages

Paragraphs (P)	1,149
Headers (H1, H2, ...)	306
Other text (TABLE, LIST, DIV, ...)	172
Java/JPQL Code (PRE)	261
Other preformatted (PRE)	84
Total Page Parts	1,972

Table 5: General Details on the Web Usage Dataset

Page views	559,852
Unique visitors (estimated)	223,268
Average Views per page	8,113.8
Average visibility time per page view	534.2 sec

Usage data were collected and recorded during a period of several months, ending in March 2020. Table 5 includes the main details about the collected usage data.

The dataset contains 559,852 views of these 69 pages (on average, 8,113.8 page views per page). To estimate unique visitors, a browser fingerprint hash was used. This provides only a rough estimate, as users with multiple computers or browsers (or who change settings) are counted more than once. The number of 223,268 unique visitors shows a low ratio of page views per visitor (about 2.5). This can be explained by the way that many visitors use this website. They arrive occasionally from a search engine to find particular information or code examples about a specific topic of JPA and leave once found what they were looking for.

On average, a page was open in the active tab of the browser for 534.2 seconds, but there is no easy way to know how long exactly the user's attention was on the page (as opposed to looking at another screen, another window, etc.). To estimate the user's attention time we need attention functions.

5.2 Attention Function Results

The proposed sub-page web analytics model is based on using an attention function. Therefore, as part of the feasibility checks, we have to see that attention functions work. Particularly, we have to check that most of the page parts can be covered by sufficient attention time, as this is essential in order to collect sufficient event data for calculating metrics and KPIs.

Subsection 4.3 defines three attention functions. Tables 6, 7, and 8 show the results of applying these three functions to the dataset. In each table, the 1,149 text paragraphs and the 261 code fragments are divided into groups by the total attention time that was allocated to each one of them, in all the dataset page views in total.

For example, with *attention3* (Table 6) only 3% of the text paragraphs and 21.8% of the code fragments received at least four hours of attention (240 minutes in the table). These page parts cover 37% and 77.1% of the total attention time, respectively. Similarly, 15.7% (8.7% + 4% + 3%) of the text paragraphs and 46% (12.3% + 11.9% + 21.8%) of the code fragments received at least one hour of attention, and these parts cover 76.6% and 93.4% of the total attention time, respectively.

The attention time that a group of parts receives could be more important than the number of these parts. For example, if we have a KPI that is expected to be effective with at least one hour of attention time, it will work for 46% of the code fragments in this dataset, but these code fragments are possibly more important than the other uncovered code fragments, as they capture 93.4% of the total attention time, according to this estimate.

Table 6: Distribution of Parts by Total *attention3* Time

Attention Time	Text (P)			Code (PRE)		
t (in minutes)	Count	%Units	%Time	Count	%Units	% Time
$0 \leq t < 5$	530	46.1%	1.6%	23	8.8%	0.1%
$5 \leq t < 15$	218	19.0%	4.8%	38	14.6%	0.8%
$15 \leq t < 30$	116	10.1%	6.1%	35	13.4%	1.5%
$30 \leq t < 60$	104	9.1%	11.0%	45	17.2%	4.1%
$60 \leq t < 120$	100	8.7%	20.8%	32	12.3%	5.6%
$120 \leq t < 240$	46	4.0%	18.8%	31	11.9%	10.7%
$240 \leq t$	35	3.0%	37.0%	57	21.8%	77.1%
Total	1,149	100%	100%	261	100%	100%

The *attention4* function (Table 7) captures and allocates more attention time than the *attention3* function (Table 6), by adding mouse move events as indicators of user attention.

Table 7: Distribution of Parts by Total *attention4* Time

Attention Time	Text (P)			Code (PRE)		
t (in minutes)	Count	%Units	%Time	Count	%Units	%Time
$0 \leq t < 5$	204	17.8%	0.2%	10	3.8%	0.0%
$5 \leq t < 15$	71	6.2%	0.3%	8	3.1%	0.1%
$15 \leq t < 30$	80	7.0%	0.9%	14	5.4%	0.3%
$30 \leq t < 60$	109	9.5%	2.4%	24	9.2%	1.2%
$60 \leq t < 120$	193	16.8%	8.3%	41	15.7%	4.1%
$120 \leq t < 240$	227	19.8%	18.8%	58	22.2%	11.5%
$240 \leq t$	265	23.1%	69.1%	106	40.6%	82.7%
Total	1,149	100%	100%	261	100%	100%

Further attention time is captured by the *attention5* function (Table 8), which allocates every second the page is open in the active tab of the browser to the visible parts. When there are no attention position indicators, attention is allocated to the visible parts in the viewport, using the normal distribution (as explained in subsection 4.3).

Table 8: Distribution of Parts by Total *attention5* Time

Attention Time	Text (P)			Code (PRE)		
t (in minutes)	Count	%Units	%Time	Count	%Units	%Time
$0 \leq t < 5$	179	15.6%	0.1%	7	2.7%	0.0%
$5 \leq t < 15$	72	6.3%	0.2%	4	1.5%	0.0%
$15 \leq t < 30$	61	5.3%	0.5%	7	2.7%	0.1%
$30 \leq t < 60$	94	8.2%	1.5%	17	6.5%	0.5%
$60 \leq t < 120$	159	13.8%	5.3%	27	10.3%	1.6%
$120 \leq t < 240$	222	19.3%	14.3%	50	19.2%	5.9%
$240 \leq t$	362	31.5%	78.1%	149	57.1%	91.9%
Total	1,149	100%	100%	261	100%	100%

In different scenarios, different attention functions may be preferred. The additional attention time that *attention5* allocates has its price, as some of the attention allocation it less accurate. The *attention3* function is the most accurate, but it has the price of losing some user activity. Choosing an attention function may be affected by various factors, including the amount of available usage data and the frequency of the events that the KPIs are based on in that data.

5.3 Event Frequency Results

Tables 9, 10, and 11 show the frequency of various events, when using *attention3*, *attention4*, and *attention5*, respectively. We can see large differences in frequencies of different events. The frequencies of some interesting events, such as Home and Search, may be too low in this dataset to be used effectively for KPIs. Other events, such as most of the mouse and scroll events, seem to have much more promising frequencies.

As expected, *attention3* produces the lowest absolute numbers of events, and *attention5* produces the highest absolute numbers. There are differences between the numbers of recorded events in the three tables, even for event types that are defined as attention position indicators themselves. The reason is that these events serve as indicators only when they occur within parts of the page content (i.e. indicating attention on a specific relevant page part), and may not be regarded as indicators by *attention3* and *attention4* if they occur outside the content, and accordingly, the attention may not be allocated.

Note that the Freq/Hour values in Tables 9, 10, and 11 can be examined against the distribution of page parts by attention time in Tables 6, 7, and 8, respectively, in order to assess how many events of each type are available in total per page part. The Freq/Hour values in Tables 9, 10, and 11, however, are not comparable among these three tables themselves, as frequencies should be examined against total attention times, which are different among these tables.

Table 9: Event Frequency With *attention3*

Event	Text (P)		Code (PRE)	
Name	Count	Per Hour	Count	Per Hour
Home	69	0.1	47	0.1
ScrollUp	639,580	938.5	776,493	987.6
ScrollDown	1,816,490	2665.3	1,815,135	2308.6
MouseMove	9,495,240	13932.4	10,447,859	13288.2
MouseOver	676	1.0	903	1.1
MouseRight	5,199,258	7628.9	5,242,925	6668.3
MouseLeft	4,286,153	6289.1	5,193,391	6605.3
Click	19,771	29.0	13,141	16.7
DblClick	45,923	67.4	26,632	33.9
Copy	4,584	6.7	35,848	45.6
Copy Word	521	0.8	507	0.6
Ctrl+F	1,341	2.0	1,716	2.2
Meta+F	238	0.3	332	0.4
Search	61	0.1	41	0.1
Blur	88,736	130.2	134,791	171.4
MouseLeave	128,595	188.7	173,985	221.3
Hide	78,753	115.6	76,404	97.2
Unload	40,256	59.1	36,289	46.2

Table 10: Event Frequency With *attention4*

Event Name	Text (P) Count	Text (P) Per Hour	Code (PRE) Count	Code (PRE) Per Hour
Home	217	0.1	87	0.1
ScrollUp	2,354,415	685.4	1,124,144	752.7
ScrollDown	8,624,083	2510.6	3,308,342	2215.1
MouseMove	24,866,954	7239.2	13,689,836	9166.0
MouseOver	676	0.2	903	0.6
MouseRight	13,049,627	3799.0	6,721,223	4500.2
MouseLeft	11,745,009	3419.2	6,956,255	4657.6
Click	31,350	9.1	10,210	6.8
DblClick	47,444	13.8	26,469	17.7
Copy	5,293	1.5	35,738	23.9
Copy Word	523	0.2	506	0.3
Ctrl+F	7,749	2.3	2,851	1.9
Meta+F	1,737	0.5	571	0.4
Search	809	0.2	1	0.0
Blur	454,855	132.4	171,020	114.5
MouseLeave	776,728	226.1	233,879	156.6
Hide	383,551	111.7	104,920	70.2
Unload	162,477	47.3	39,278	26.3

Table 11: Event Frequency With *attention5*

Event Name	Text (P) Count	Text (P) Per Hour	Code (PRE) Count	Code (PRE) Per Hour
Home	306	0.1	125	0.1
ScrollUp	2,901,207	639.4	1,693,032	686.1
ScrollDown	10,898,916	2402.2	6,147,258	2491.3
MouseMove	25,361,597	5589.8	13,963,012	5658.9
MouseOver	689	0.2	921	0.4
MouseRight	13,309,173	2933.4	6,855,331	2778.3
MouseLeft	11,978,669	2640.2	7,095,083	2875.5
Click	31,935	7.0	10,415	4.2
DblClick	48,415	10.7	27,016	10.9
Copy	5,396	1.2	36,451	14.8
CopyWord	533	0.1	516	0.2
Ctrl+F	8,343	1.8	3,647	1.5
Meta+F	2,049	0.5	913	0.4
Search	823	0.2	1	0.0
Blur	571,243	125.9	251,878	102.1
MouseLeave	801,682	176.7	251,461	101.9
Hide	504,855	111.3	181,306	73.5
Unload	203,007	44.7	64,198	26.0

A new event, CopyWord, appears in Tables 9, 10, and 11. This event is extracted from the Copy event by selecting only the copy operations of individual lower case words. Although the frequency of this new event seems quite low, so its effectiveness is expected to be limited, subsection 5.5 demonstrates that it could still serve as a valuable KPI, at least to some extent.

5.4 A Positive KPI Example

As described in subsection 5.1, many visitors to this website arrive from search engines, looking for code examples of a specific topic of JPA. When a requested code example is found, a common user action is to copy relevant code to the clipboard in order to paste it later in an IDE [21]. This activity is in line with the objective of the website to serve as a source of technical information. Therefore, we can consider the $eventFreq(Copy, p)$ metric, in code fragments, as a positive KPI (in short, the Copy KPI), because a high frequency of this event is a good indication.

For the Copy KPI, the three attention functions are almost equivalent. The reason for this is that the vast majority of copy operations start with a mouse click to select text or code for copying. A mouse click is considered as an attention position indicator by all the three attention functions, so the attention is captured and most Copy events are never missed. The results in this section, however, are based on the *attention3* function.

Figures 3, 4, and 5 show the code fragments with the highest $eventFreq(Copy, p)$ values among all the code fragments with at least one hour of total attention time (fragments that cover 93.4% of the user's attention time, according to Table 6).

```
SELECT c.name FROM Country c
WHERE c.population > 1000000 ORDER BY c.name
```

Figure 3: Copying From a Query (Copy = 155.5 e/h)

```
Class ProjectId {
    int departmentId;
    long projectId;
}
```

Figure 4: Copying an ID Class (Copy = 146.2 e/h)

```
@Entity
@Index(members={"lastName","firstName"})
public class EntityWithCompositeIndex {
    String firstName;
    String lastName;
}
```

Figure 5: Copying an Index Annotation (Copy = 136.2 e/h)

As Table 9 shows, the average frequency of the Copy event in code fragments is **45.6** events per attention hour (e/h). The values for the code fragments in Figures 3, 4, and 5 are **155.5**, **146.2**, and **136.2** events per attention hour, respectively. This is a good indication (although not a decisive proof) that users find these code fragments useful.

The green frames in these figures mark the most commonly copied code from these code fragments. The string "ORDER BY c.name", in Figure 3, was copied 88 times. It shows that the users identify the important parts in this code example, which focuses on using ORDER BY in queries. In Figure 4, the entire code was copied 131 times. In Figure 5, the marked index definition, which is the core of this example, was copied 211 times. These examples show that web analytics may be possible, even for smaller parts than paragraphs and code fragments, e.g. for groups of words in a code fragment.

5.5 A Negative KPI Example

Although positive KPIs are useful, negative KPIs could be even more useful, as they can help in spotting issues that require improvements to the website. This subsection demonstrates this capability with a variant of the Copy metric, the CopyWord metric, as a negative KPI. The CopyWord event refers to copy operations of a single word. A word is defined in this context as a string that consists of lower case letters only. This simplified but strict definition of words increases the precision (i.e. code element names, which are not real words, are excluded), at the cost of losing some legitimate words, which is acceptable for the purpose of this demonstration.

The motivation behind focusing on the CopyWord event is that users copy single words in the text to the clipboard when the meaning of these words is not completely clear to them, in order to search for further information on the internet (e.g in dictionaries). Therefore, a high frequency of CopyWord events in a specific paragraph may indicate that the text is relatively complicated to understand by the audience of the website.

Figure 6 shows the paragraph with the highest CopyWord frequency in the dataset, **10.1** CopyWord events per attention hour, among all the text paragraphs with at least one hour of total attention time (covering 76.6% of the total user attention time, according to Table 6). Note that the average frequency of CopyWord events, according to Table 9, is just **0.8** per attention hour.

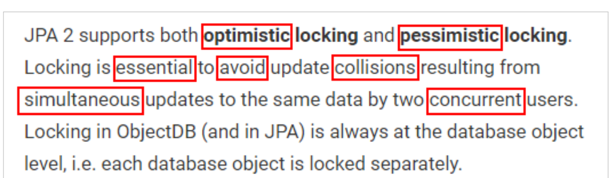
JPA 2 supports both **optimistic** **locking** and **pessimistic** **locking**. Locking is essential to avoid update collisions resulting from simultaneous updates to the same data by two concurrent users. Locking in ObjectDB (and in JPA) is always at the database object level, i.e. each database object is locked separately.

Figure 6: The Paragraph With the Highest CopyWord Value (10.1 e/h)

The red frames in Figure 6 mark the words that were copied by users. Most of the marked words do seem advanced (at least for non-native English speakers) and less commonly used, compared to the words that were not copied. We can assume that less commonly used words are more likely to challenge users. Table 12 shows the relative frequency of these words, based on a list of 20,000 words, the "20k list", which was produced from the Google's Trillion Word Corpus [8].

Table 12: Copied Words in Figure 6

Copied Word	Times Copied	Word Rank by the 20k List: # in 20k List	# in Website
pessimistic	4	N/A	N/A
simultaneous	2	11,933 / 20,000	2,821 / 3,134
collisions	2	18,134 / 20,000	3,079 / 3,134
concurrent	1	10,633 / 20,000	2,748 / 3,134
essential	1	2,131 / 20,000	1,242 / 3,134
optimistic	1	14,413 / 20,000	2,964 / 3,134
avoid	1	2,215 / 20,000	1,277 / 3,134

The 20k list is ordered in descending frequency, starting from the most frequent words ("the", "of"). So the positions of words in the list provide ranking by frequency. The examined web pages include 3,134 different words from the 20k list, as well as some words that are not on the list. We also use the 20k list to rank the sublist of 3,134 words that appear in the examined web pages.

The word that was copied the most, "pessimistic", is not commonly used, as it is not even on the 20k list. Other words, such as "simultaneous", "collisions", "concurrent", and "optimistic" are on the 20k list, but in low positions, indicating that they are also not very common. The word "simultaneous", for example, is ranked as #11,933 on the 20k list and #2,821 on the sublist of 3,134 website words. The first 10,000 words on the 20k list cover 96.9% of the occurrences of words in the examined web pages, as shown in Table 13, so it is easy to see that this paragraph has a high concentration of less frequently used words.

Table 13: Coverage of the Website Words by the 20k List

Words with Position *pos* in the 20k List	This Group Count	This Group Share	Accumulated Count	Accumulated Share
$0 < pos \leq 1,000$	60,328	67.0%	60,328	67.0%
$1,000 < pos \leq 2,000$	9,330	10.4%	69,658	77.4%
$2,000 < pos \leq 3,000$	5,845	6.5%	75,503	83.9%
$3,000 < pos \leq 4,000$	3,329	3.7%	78,832	87.6%
$4,000 < pos \leq 5,000$	2,971	3.3%	81,803	90.9%
$5,000 < pos \leq 6,000$	1,383	1.5%	83,186	92.4%
$6,000 < pos \leq 7,000$	1,194	1.3%	84,380	93.8%
$7,000 < pos \leq 8,000$	922	1.0%	85,302	94.8%
$8,000 < pos \leq 9,000$	931	1.0%	86,233	95.8%
$9,000 < pos \leq 10,000$	981	1.1%	87,214	96.9%
$10,000 < pos \leq 11,000$	265	0.3%	87,479	97.2%
$11,000 < pos \leq 12,000$	235	0.3%	87,714	97.5%
$12,000 < pos \leq 13,000$	428	0.5%	88,142	97.9%
$13,000 < pos \leq 14,000$	313	0.3%	88,455	98.3%
$14,000 < pos \leq 15,000$	598	0.7%	89,053	98.9%
$15,000 < pos \leq 16,000$	86	0.1%	89,139	99.0%
$16,000 < pos \leq 17,000$	245	0.3%	89,384	99.3%
$17,000 < pos \leq 18,000$	271	0.3%	89,655	99.6%
$18,000 < pos \leq 19,000$	296	0.3%	89,951	99.9%
$19,000 < pos \leq 20,000$	50	0.1%	90,001	100.0%

Figure 7 shows another example, the paragraph with the second highest CopyWord frequency, **9.2** CopyWord events per attention hour.

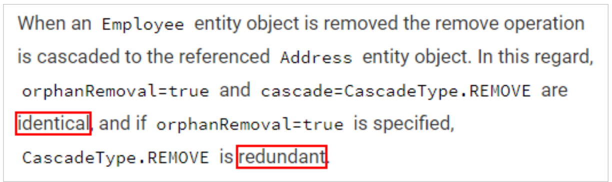
When an `Employee` entity object is removed the remove operation is cascaded to the referenced `Address` entity object. In this regard, `orphanRemoval=true` and `cascade=CascadeType.REMOVE` are identical, and if `orphanRemoval=true` is specified, `CascadeType.REMOVE` is redundant.

Figure 7: The Paragraph With the 2nd Highest CopyWord Value (9.2 e/h)

In this example, there is one dominant infrequent word, "redundant" (see Table 14), which was copied 15 times.

Table 14: Copied Words in Figure 7

Copied Word	Times Copied	Word Rank by the 20k List	
		# in 20k List	# in Website
redundant	15	12,423 / 20,000	2,850 / 3,134
identical	2	6,552 / 20,000	2,299 / 3,134

In both examples, it may be possible to rephrase the text in a way that will make it easier to read. Therefore, the CopyWord negative KPI seems to be useful in spotting paragraphs that may require rephrasing, for the benefit of users with elementary proficiency in English.

Note that the infrequency of words is not the only factor that makes sentences more challenging to understand. The context in which an infrequent word appears may also affect the readability and the understandability of the sentence. In many cases, readers can understand the main ideas in the text without a full understanding of every word. Figure 7, however, demonstrates a situation in which the main challenging word, "redundant", plays a central role, and it is impossible to understand this statement without knowing the meaning of that word. Therefore, the CopyWord KPI does more than just highlighting infrequent words in the text. It provides real, implicit feedback from users on places in the text in which infrequent words are more challenging, as in this example.

The CopyWord KPI demonstrates that even a low-frequency metric (0.8 copies per attention hour) can be useful as a KPI, and it shows that even a relatively small amount of data could be useful in sub-page web analytics. However, a low-frequency limits the extent to which events can be used, and in this dataset, the CopyWord KPI, due to its low-frequency, is only effective for paragraphs with sufficient user attention time. Finding KPIs that are based on more frequent events (e.g. scroll and mouse movement) could extend the scope in which sub-page web analytics could be effective.

6 CONCLUSIONS

This study introduced the idea of sub-page web analytics and sub-page metrics and KPIs. We discussed the potential and the challenges, proposed a model for defining and calculating sub-page metrics and KPIs, and examined sub-page web analytics concepts and methods using real web usage data from a technical-educational website.

The purpose of sub-page web analytics is to extend and refine the knowledge that conventional web analytics generates. As with satellite images, which can expose more details when the resolution is increased, the motivation behind sub-page web analytics is to increase the resolution of the web usage image that existing web analytics tools provide, exposing new details on websites and user behavior, which are currently hidden.

The preliminary results are very encouraging. The experimental part of this study demonstrates two types of KPIs that are based on copy-to-clipboard operations of website visitors: a positive KPI and a negative KPI. The positive KPI highlights code examples on the website that are likely to be useful to the website users. The negative KPI exposes paragraphs of text that apparently some users find more challenging to understand.

This study is the first step towards extending web-analytics, metrics, and KPIs from website and page levels to the sub-page level. Future work may include the examination of these concepts and methods with different types of websites and exploring and studying additional metrics and KPIs.

REFERENCES

[1] Feras Alhlou, Shiraz Asif, and Eric Fettman. 2016. *Google Analytics Breakthrough: From Zero to Business Impact.* John Wiley & Sons, USA.

[2] 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 (ETRA '12).* Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/2168556.2168575

[3] Dave Chaffey and Mark Patron. 2012. From web analytics to digital marketing optimization: Increasing the commercial value of digital analytics. *Journal of Direct, Data and Digital Marketing Practice* 14 (07 2012), 30–45. https://doi.org/10.1057/dddmp.2012.20

[4] Mark Claypool, Phong Le, Makoto Wased, and David Brown. 2001. Implicit Interest Indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI '01).* Association for Computing Machinery, New York, NY, USA, 33–40. https://doi.org/10.1145/359784.359836

[5] Matthew Conlen, Alex Kale, and Jeffrey Heer. 2019. Capture & Analysis of Active Reading Behaviors for Interactive Articles on the Web. *Computer Graphics Forum* 38, 3 (2019), 687–698. https://doi.org/10.1111/cgf.13720 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13720

[6] Datanyze. 2020. *Web Analytics Market Share Report.* Datanyze. https://www.datanyze.com/market-share/web-analytics--1

[7] Brent Dykes. 2014. *Web Analytics Kick Start Guide: A Primer on the Fundamentals of Digital Analytics.* Peachpit, Pearson Education, USA.

[8] first20hours. 2020. *20k.txt, most common English words in order of frequency, as determined by n-gram frequency analysis of the Google's Trillion Word Corpus.* GitHub. https://github.com/first20hours/google-10000-english

[9] Max Grusky, Jeiran Jahani, Josh Schwartz, Dan Valente, Yoav Artzi, and Mor Naaman. 2017. Modeling Sub-Document Attention Using Viewport Time. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17).* Association for Computing Machinery, New York, NY, USA, 6475–6480. https://doi.org/10.1145/3025453.3025916

[10] Qi Guo and Eugene Agichtein. 2008. Exploring Mouse Movements for Inferring Query Intent. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08).* Association for Computing Machinery, New York, NY, USA, 707–708. https://doi.org/10.1145/1390334.1390462

[11] Qi Guo and Eugene Agichtein. 2012. Beyond Dwell Time: Estimating Document Relevance from Cursor Movements and Other Post-Click Searcher Behavior. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12).* Association for Computing Machinery, New York, NY, USA, 569–578. https://doi.org/10.1145/2187836.2187914

[12] Layla Hasan, Anne Morris, and Steve Probets. 2009. Using Google Analytics to Evaluate the Usability of E-Commerce Sites. In *Proceedings of the 1st International Conference on Human Centered Design.* Springer Berlin Heidelberg, Berlin, Heidelberg, 697–706. https://doi.org/10.1007/978-3-642-02806-9_81

[13] David Hauger, Alexandros Paramythis, and Stephan Weibelzahl. 2011. Using Browser Interaction Data to Determine Page Reading Behavior. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization (UMAP'11)*. Springer-Verlag, Berlin, Heidelberg, 147–158.

[14] Jeff Huang, Ryen White, and Georg Buscher. 2012. User See, User Point: Gaze and Cursor Alignment in Web Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1341–1350. https://doi.org/10.1145/2207676.2208591

[15] Shamsi Iqbal and Brian Bailey. 2004. Using Eye Gaze Patterns to Identify User Tasks. *The Grace Hopper Celebration of Women in Computing* 04 (01 2004), 6.

[16] Edson C. Tandoc Jr. 2015. Why Web Analytics Click. *Journalism Studies* 16, 6 (2015), 782–799. https://doi.org/10.1080/1461670X.2014.946309 arXiv:https://doi.org/10.1080/1461670X.2014.946309

[17] Joel Järvinen and Heikki Karjaluoto. 2015. The use of Web analytics for digital marketing performance measurement. *Industrial Marketing Management* 50 (2015), 117 – 127. https://doi.org/10.1016/j.indmarman.2015.04.009

[18] Avinash Kaushik. 2007. *Web Analytics: An Hour a Day*. SYBEX Inc., USA.

[19] Avinash Kaushik. 2010. *Web Analytics 2.0*. SYBEX Inc., USA.

[20] Ilan Kirsh. 2020. Directions and Speeds of Mouse Movements on a Website and Reading Patterns: A Web Usage Mining Case Study. In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020), Biarritz, France*. Association for Computing Machinery, New York, NY, USA, 129–138. https://doi.org/10.1145/3405962.3405982

[21] Ilan Kirsh and Mike Joy. 2020. A Different Web Analytics Perspective Through Copy to Clipboard Heatmaps. In *Proceedings of the 20th International Conference on Web Engineering (ICWE 2020), Lecture Notes in Computer Science, vol 12128*. Springer International Publishing, Cham, 543–546. https://doi.org/10.1007/978-3-030-50578-3_41

[22] Ilan Kirsh, Mike Joy, and Yoram Kirsh. 2020. Horizontal Mouse Movements (HMMs) on Web Pages as Indicators of User Interest. In *Proceedings of the 22nd HCI International Conference (HCII 2020), Communications in Computer and Information Science*. Springer International Publishing, Cham, 8.

[23] Dmitry Lagun and Mounia Lalmas. 2016. Understanding User Attention and Engagement in Online News Reading. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, New York, NY, USA, 113–122. https://doi.org/10.1145/2835776.2835833

[24] Fabrizio Lamberti and Gianluca Paravati. 2015. VDHM: Viewport-DOM Based Heat Maps as a Tool for Visually Aggregating Web Users' Interaction Data from Mobile and Heterogeneous Devices. In *Proceedings of the 2015 IEEE International Conference on Mobile Services (MS '15)*. IEEE Computer Society, USA, 33–40. https://doi.org/10.1109/MobServ.2015.15

[25] F. Lamberti, Gianluca Paravati, Valentina Gatteschi, and Alberto Cannavò. 2017. Supporting Web Analytics by Aggregating User Interaction Data From Heterogeneous Devices Using Viewport-DOM-Based Heat Maps. *IEEE Transactions on Industrial Informatics* 13 (08 2017), 1989 – 1999. Issue 4. https://doi.org/10.1109/TII.2017.2658663

[26] JooWon Lee and Jae-Hyeon Ahn. 2012. Attention to Banner Ads and Their Effectiveness: An Eye-Tracking Approach. *International Journal of Electronic Commerce* 17, 1 (2012), 119–137. https://doi.org/10.2753/JEC1086-4415170105 arXiv:https://doi.org/10.2753/JEC1086-4415170105

[27] Yixuan Li, Pingmei Xu, Dmitry Lagun, and Vidhya Navalpakkam. 2017. Towards Measuring and Inferring User Interest from Gaze. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 525–533. https://doi.org/10.1145/3041021.3054182

[28] Heng Luo, Stevie Rocco, and Carl Schaad. 2015. Using Google Analytics to Understand Online Learning: A Case Study of a Graduate-Level Online Course. In *Proceedings of the 2015 International Conference of Educational Innovation through Technology (EITT '15)*. IEEE Computer Society, USA, 264–268. https://doi.org/10.1109/EITT.2015.62

[29] Alexandre Milisavljevic, Kevin Hamard, Coralie Petermann, Bernard Gosselin, Karine Doré-Mazars, and Matei Mancas. 2018. Eye and Mouse Coordination During Task: From Behaviour to Prediction. In *International Conference on Human Computer Interaction Theory and Applications*. SciTePress, Setúbal, Portugal, 86–93. https://doi.org/10.5220/0006618800860093

[30] Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex Smola. 2013. Measurement and Modeling of Eye-Mouse Behavior in the Presence of Nonlinear Page Layouts. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*. Association for Computing Machinery, New York, NY, USA, 953–964. https://doi.org/10.1145/2488388.2488471

[31] Kerry Rodden and Xin Fu. 2007. Exploring How Mouse Movements Relate to Eye Movements on Web Search Results Pages. In *Proceedings of ACM SIGIR 2007 Workshop on Web Information Seeking and Interaction*. Association for Computing Machinery, New York, NY, USA, 29–32. http://research.microsoft.com/~ryenw/proceedings/WISI2007.pdf

[32] Uzi Smadja, Max Grusky, Yoav Artzi, and Mor Naaman. 2019. Understanding Reader Backtracking Behavior in Online News Articles. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3237–3243. https://doi.org/10.1145/3308558.3313571

[33] Songhua Xu, Hao Jiang, and Francis C.M. Lau. 2009. User-Oriented Document Summarization through Vision-Based Eye-Tracking. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. Association for Computing Machinery, New York, NY, USA, 7–16. https://doi.org/10.1145/1502650.1502656