Your eyes explain everything: exploring the use of eye tracking to provide explanations on-the-fly

Martijn Millecamp[1], Toon Willemot[1] and Katrien Verbert[1]

[1]*Departement of Computer Science KU Leuven, Celestijnenlaan 200 A bus 2402, 3001 Heverlee, Belgium*

Abstract
Despite the proven advantages, explanations are not yet mainstream in industry applications of recommender systems. One of the possible reasons for this lack of adaption is the risk of overwhelming end-users with the explanations. In this paper, we investigate whether it is possible to overcome the information overload problem by only showing explanations that are relevant to the user. To do so, we leverage the gaze of a user as a novel responsiveness technique. We first conducted a co-design session to discuss several design decisions of a gaze responsive music recommender interface. As a next step, we implemented a gaze responsive music recommender interface and compared it in a between-subject user study (N=46) to two interfaces with more classical responsive techniques: a hover responsive interface and a click responsive interface. Our results show that providing explanations based on gaze is a promising solution to provide explanations on-the-fly.

Keywords
Eye tracking, Explanations, Music recommender systems, User studies

1. Introduction

By providing personalized items to users, recommender systems help users to find items that fit their needs out of an abundance of options [1]. Several studies have highlighted the key role of explaining recommendations to end-users as a basis to increase user trust and acceptance of recommendations [2, 3, 4]. However, it has also been shown that providing explanations also involves risks [5]. For example, explanation could overwhelm the users by showing too much information [5, 6].

A possible solution to overcome this increased information load could be to provide the user with control over the visibility of explanations. However, providing such control is challenging, as several studies showed that there is a risk that users do not use or stop using such controls because it is too demanding [7]. Therefore, we investigate if providing explanations based on gaze data can decrease the effort to ask for explanations, as several studies have already shown that using the gaze of a user to interact with an interface is perceived as easier and more efficient than with a mouse [8].

More concretely, in this study we provide explanations on-the-fly in a music recommender system based on the recommendation on which the user is looking at. Additionally, we investigate whether the way of providing explanations would compromise the user experience in comparison with more traditional methods such as providing explanations on click or on hover.

As far as we know, this study is the first to use gaze to interact with explanations for recommender system. We started with a co-design session to discuss several design decisions involved in creating a responsive music recommender system interface. Based on the results of this co-design session, we implemented three different music recommender system interfaces: one which shows explanation after clicking on a button (Click), one which shows explanations when the user hovers over a recommendation (Hover) and one in which the explanation is shown when the user is focusing on a recommended song (Gaze). All three interfaces used the Spotify API to generate recommendations. In a between-subject study (N=46), we measured the *usability, use intention, satisfaction, information sufficiency* and *decision support* of the different interfaces. Our results show that a gaze responsive interface is a promising solution for dynamically providing explanation to avoid information overload.

2. Related work

2.1. Eye tracking

Today, most of the popular device designs such as smartphones and laptops have high-quality, user-facing cameras. As a consequence, gaze tracking using these cameras will become easier and will be increasingly used as interaction method in the future [9]. Several researchers in HCI already successfully explored the use of eye tracking as input device [10]. For example, Shakil et al. [9] implemented CodeGazer, a system to navigate through source code by using gaze. They showed that users liked and even preferred this gaze based navigation over traditional interactions.

In the next paragraphs, we will provide an overview of different methods to integrate gaze as an interaction method. Following the taxonomy of Lutteroth et al. [8], we divided these interaction methods in three different categories: direct, indirect and auxiliary.

2.1.1. Direct

A first possibility to use eye tracking as input, is by using the point of focus directly to trigger an action. There are several options to trigger the action on which the user is focusing such as blinks, winks and eyebrow movement, but the most used action is just focusing on the responsive element for a longer time [8]. The disadvantage of this last method is that it can trigger undesired actions if the threshold focus time is not long enough or due to involuntary eye movements [11]. However, the alternative methods often suffer also from involuntary movements or become less efficient than clicking [12, 8].

The biggest limitation of using gaze directly as interaction method, is that the accuracy of eye tracking needs to be high enough to avoid triggering the incorrect action [13]. At this moment, the accuracy of eye trackers is often not high enough to use this method without modifying the interface (e.g. enlarging all interaction elements) [13]. Several studies have already proposed

a variety of magnification techniques to overcome this problem [8]. For example, the ERICA system solves this accuracy problem by magnifying a region of interest when the user dwells long enough on this area [14]. Ashmore et al. [15] used a different approach by using a fish eye lens. They found that a dwell-activated fish-eye lens works better than a continuous fish-eye zoom.

In this study, we will use gaze as a direct interaction method. Nonetheless, in contrast to the methods described above, we will not use gaze as an alternative to a click but only as an additional interaction method to show explanations. We will discuss this in more detail in Section 4.

2.1.2. Indirect

A second possibility to use eye tracking as input, is by providing additional selection elements that can help to distinguish on which target the user wants to click [8]. An example of this are confirm buttons as implemented in the study of Penkar et al. [16]. Every time they detected that the user was focusing for a longer period in the same area, they created for each interaction element in that area a larger button. By focusing on one of these buttons, users could confirm which action they wanted to trigger. The study of Lutteroth et al. [8] built onto this idea, but they colored the interaction elements and instead of generating buttons on-the-fly they provided a fixed side bar with colored buttons to confirm the element. Although this method was not yet able to be more efficient than the mouse, users perceived this method as faster [8].

2.1.3. Auxiliary

A third possibility is not to use gaze as an interaction mechanism, but as a way to speed up the mouse movements. For example, the study of Zhai et al. [17] used gaze to quickly move the mouse to the focus point after which the user could use the mouse for selecting the correct interaction element. Similarly, the study of Blanch and Ortega [18] first used eye tracking to move the mouse to a cell in a grid in which the user wants to interact, after which the user could take over control and use the mouse for triggering the action.

2.2. Explanations

Due to the increasing popularity of recommender systems and the number of decisions we base on these systems, there is also a growing amount of concern about the black box nature of these systems [19, 20]. One of the possibilities to open this black box to the users, is by providing explanations to the users [4]. These explanations could explain to the user why the system recommends a specific item or even achieve that the user has a causal understanding of why the item is recommended [21]. Moreover, providing explanations could not only increase transparency and trust, but they could also help to increase efficiency, user satisfaction, effectiveness or even help to persuade a user to consume an item.

Despite these many advantages, providing explanations can also have some risks such as over-trust, under-trust, suspicious motivation and information overload [5]. In this study we will focus especially on information overload which can happen when there is too much information given to the user or when the explanations are too complex [22, 5]. In the study of Kulesza

et al. [23], they found that the most sound and most complete explanations help users the best to understand the recommender system. However, they also argue that providing all this information comes at a cost. In this study, we want to prevent information overload caused by explanations by providing users only the explanation of the item at which they are looking.

3. Co-design session

As there is not yet a vast amount of research about the use of gaze in recommender system interfaces to trigger information, we started with a co-design session to gather user input about several design decisions such as the way gaze can be used, what information needs to be shown and where this information should appear.

To involve both experts and non-experts, we recruited three design experts who are active as front-end developers and three students (1F) without design expertise who regularly use music streaming services. To make sure all participants were familiar with the possibilities of the Spotify API, we distributed a hand-out to all users on which we listed which information could be provided about a song by the Spotify API. Next, the users were split into two groups: the group of experts and the group of consumers. In each group, users were given the task to design together a gaze-responsive music recommender interface. Afterwards, a group discussion was held to determine how gaze we could be used in a responsive music recommender system. In the next paragraph, we will discuss the main results of this group discussion.

3.1. Results

(In)direct use of gaze: The first point we discussed was how the gaze would be used: direct, indirect or auxiliary (see Section 2.1).

In the discussion, the initial proposal was to use gaze indirectly by showing a confirmation icon next to the focus point whenever the user was looking to an element that was responsive. By using a second dwell, users could then trigger the action. However, it was argued that this approach was again expecting from the user to actively demand explanations. To overcome this limitation, the second proposal was to use dwell directly as a trigger to show relevant information to the user. In this proposal, users would not need to explicitly demand for information which will feel more natural and less demanding than indirect use of gaze. The participants argued that the disadvantage of this proposal was that it could lead to inadvertent triggering of information, but the main opinion in the discussion was that this appearance of undesired information would not distract the user. In this study, we choose to implement the use of gaze directly to trigger explanations.

Gaze responsive elements: Another point of discussion was about which elements in the interface would be responsive and what actions they should trigger. At the end of the discussion, everyone agreed that it would be useful to use gaze to show explanations only when the user focuses on a recommended song. They also agreed that this extra information should best appear in a separate non-responsive area to avoid changes in the interface when they are reading this extra information. Additionally, to avoid too much distraction, this extra information should

appear on the right side of the screen and this information cannot appear too abruptly as this would draw too much attention. The suggestion to user a smoother transition such as fade-in was implemented.

4. Interface

4.1. General interface

As shown in Figure 1, the implemented interface consist of four different elements. In the top left corner (Part A), users can modify different audio features to steer the recommendation process. Underneath (Part B), users can see the songs they picked as seeds in the initial phase and change these if they want to. In the central column (Part C), users can see a list of recommendations displayed by title, artist and the cover of the album. Additionally, they can listen to a 30 second preview of the song by clicking on the play button and add as song to their playlist by clicking on the heart icon. All songs in Part B and C are responsive and when triggered the explanation and some additional information about that song is shown in Part D. This part is not gaze responsive and shows additional information about the selected song such as the duration of the song, the popularity, the evolution of the loudness and whether or not the song contains explicit lyrics. Underneath, it is explained to the users that the song is recommended because it has similar features as their own preference given in Part A.

In the study procedure, before users entered this main interface of the application, users were asked to select up to five songs they liked. These songs were then used as seeds to retrieve recommendations from the Spotify API. [1] Additionally, the user can adjust three different audio features to steer the recommendation process. These audio features were *Danceability*, *Energy* and *Tempo*. [2] When the users were happy with their choice, they continued to the main interface of the application described above.

4.2. Differences between the interfaces

To benchmark the gaze responsive interface against more traditional approaches, we implemented three different interfaces: one with explanations on click (Click), one with explanations when the user hovers over the recommendation (Hover) and one in which the explanation was shown when the user focused on the recommended song (Gaze).

For Gaze, the extra information about a recommended song is triggered directly without a confirmation action. When we detected a fixation that lasted longer than 300 ms and that was located on a song in the responsive areas (Part B and Part C in Figure 1) we started to fade-in the explanation and some additional information about that song in Part D of Figure 1. To avoid inadvertent actions, we made sure that the recommended songs are large enough. Additionally, to avoid distraction when the explanations appear, we made the appearance of information as smooth as possible through a slow fade-in.

For both Hover and Click, we used the same interface as Gaze. To enable fair comparison, Hover used the same threshold of 300ms before information about a song started to fade-in,

---

[1]https://developer.spotify.com/documentation/web-api/reference/browse/get-recommendations/

[2]https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

Figure 1: A screenshot of the gaze-responsive interface with the responsive areas indicated in blue. A: Audio features, B: Seeds, C: Recommendations, D: Details of the song

but in this interface it was the threshold of hovering over a song instead of fixating on it. As mentioned before, for Click, users needed to click on a song to see the additional information.

4.3. Eye tracking

To capture the gaze of the user, we used a remote eye tracker, namely the Tobii 4C. This eye tracker has a sampling rate of 90 Hz and comes with its own calibration software. We implemented the IV-T algorithm to be able to detect fixations from the raw gaze data in real time [24]. This algorithm needs only one parameter, namely the angle velocity threshold, which was set to 20 degrees per second based on the study of Sen and Megaw [25]. As part of the contribution of this paper, all of the code is open-source and online available. [3]

5. Methodology

To investigate whether the experience of users with a gaze responsive music recommender interface is similar to more traditional responsiveness techniques, we conducted a between-subject study in which we measured the user experience of three different interfaces (Gaze, Hover, and Click) and several gaze responsive aspects of Gaze. The participants, the study procedure as well as the measurements are described in detail below.

---

[3]https://github.com/WToon/thesis-frontend https://github.com/WToon/thesis-backend

5.1. Participants

In total, 46 participants were recruited for this study through e-mailing lists or social media. A total of 17 users (5 Females) tested the gaze responsive interface, a total of 14 users (5 Females) tested the mouse responsive interface and 15 users (3 Females) tested the click responsive interface. Thirty-four users were between 18 and 24, eight users between 25 and 34, two users between 35 and 44 and two others were 45 or older.

5.2. Study Procedure

Due to the COVID-19 guidelines, we minimized the face-to-face contact moments by conducting the study with Click and Hover online. Only the evaluation of Gaze could not be held online because of the use of the eye tracker.

However, all experiments followed the same procedure:
The experiment started with an initial phase in which users filled in an informed consent form and a questionnaire to gather information regarding their age and gender. Afterwards, users were given the task to create a playlist of five songs which they would listen to when they are in a happy mood. Afterwards, they were directed to the start screen which is described in Section 4. When users were finished with creating their playlist of five songs, they were asked to fill in a questionnaire which will be discussed in the next Section 5.3.

For the users who tested the gaze responsive interface, there was an additional calibration phase. After the initialization phase, we calibrated the eye tracker to the user using the Tobii 4C calibration software.

5.3. Measurements

5.3.1. User experience

We measured user experience via four subjective system aspects described by Knijnenburg et al. [26] including *use intention: I would use this application again*, *satisfaction: Overall, I am satisfied with this application*, *information sufficiency: The interface provided me enough information about the recommendations* and *decision support: The extra information helped me to make a decision*. To measure these aspects, we asked users to rate these four questions on a 5-points Likert scale.
Additionally, we used the SUS-questionnaire to compare the *usability* of the different interfaces [27].

5.3.2. Gaze responsive aspects

The users who tested the gaze responsive interface were also asked to rate on a 5-point Likert scale three more questions about *intrusiveness: The appearance of the data is too intrusive* , *accuracy: The eye tracker is accurate* and *activation time: the information is shown too quickly.*
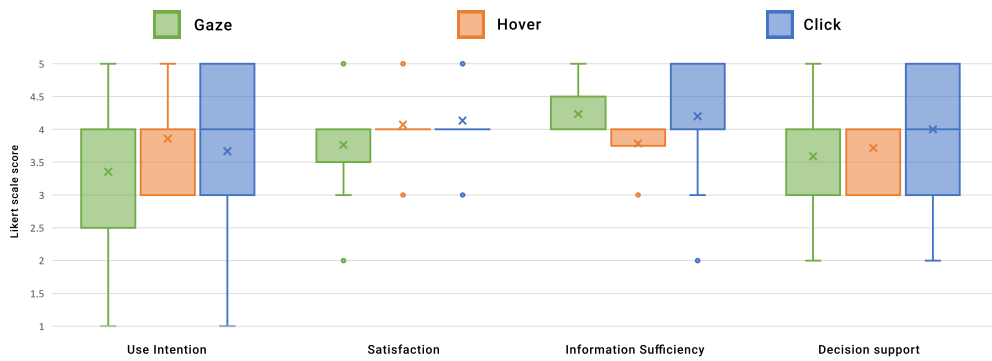
Figure 2: The results of the subject system aspects for the different interfaces: the gaze responsive interface (Gaze-green), the mouse responsive interface (Hover-orange) and the information on demand interface (Click-blue)

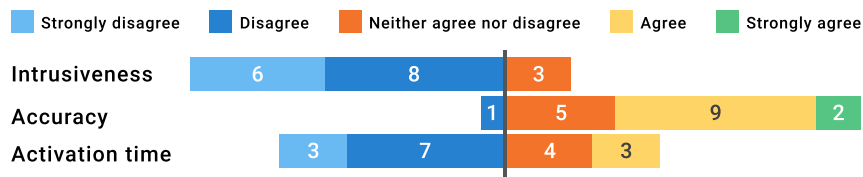# 6. Results

6.1. User experience

SUS To test the usability of the interfaces, we asked all participants to fill in the SUS-questionnaire [27]. All interfaces reached a score between 72 and 85 (Gaze: 77.5 ± 14.51, Hover: 81.6 ± 8.69, Click: 79.5 ± 8.30) which is considered between good and excellent usability [27]. We expected to see a significant decrease in usability as eye tracking is a new technology and clicking is the gold standard. However, a Kruskal-Wallis H test did not reveal significant differences (*H= 1.202, p=.548*).

Subjective system aspects. Next to usability, we also measured four subjective system aspects for each interface. The results of these different aspects are shown in Figure 2. On this figure we can see that the gaze responsive interface scored lower than the other two interfaces for *use intention*, *satisfaction* and *decision support*. For *information sufficiency* the gaze responsive interface scored better than the mouse responsive interface, and even slightly better than the information on demand interface. However, a Kruskal-Wallis H test did not reveal significant differences between the different interfaces.

6.2. Gaze responsive aspects

As discussed in Section 5, participants who tested the gaze responsive interface also rated three different aspect specific of the gaze responsiveness. We asked participants whether the appearance of the data was too intrusive, but none of the participants found the data appearance intrusive as shown in Figure 3. We also asked whether the eye tracker was accurate and eleven users reported that the eye tracker was accurate, one participant reported that the eye tracker was not accurate and five users neither agreed or disagreed. In the last question we asked users whether showing the information after 300 ms was not too quickly and the results show that ten of the users found 300 ms a good timing, but also that seven users might prefer to see the

Figure 3: The results of the gaze responsive system aspects

information after a longer time threshold.

7. Discussion

As mentioned in Section 4.2, we implemented a fade-in of information after 300 ms. The motivation behind this smooth transition was to not distract users when inadvertent explanations were shown. As shown in Figure 3, no users reported that the transition of information was too obtrusive which suggest that this is a good solution. Figure 3 also shows that fading-in information after 300 ms might be a little bit too quickly as only ten users agreed that 300 ms was not too quickly.

As shown in Figure 3, the eye tracker was not considered accurate for six out of the seventeen participants. This is not completely unexpected as we used the Tobii 4C eye tracker which is mostly designed for gaming and as such does not reach the same accuracy as more advanced and more expensive models. Surprisingly, this did only lead to a small loss in usability which was not found to be significant as described in Section 6. A possible explanation for this might be that gaze is both fast, natural and thus usable even as the accuracy is sometimes limited [8]. Moreover, this natural feeling of leveraging gaze to fade-in information might also explain the trend why the *information sufficiency* is the highest for the gaze responsive interface. We assume that for *information sufficiency*, the natural appearance of information counterbalanced the accuracy limitations. For the other subjective system aspects we see only an insignificant trend that the gaze responsive interface scored lower.

As we did not use the most advanced eye tracker and as we expect that advantages in eye tracking technology will further increase the accuracy, we argue that using gaze to show explanations on-the-fly is a promising direction to avoid information overload in recommender systems interfaces.

8. Conclusion

In this paper we explored the use of gaze in a music recommender system interface to dynamically show explanations based on where the user is looking. Based on the results of a co-design session, we implemented a gaze responsive interface (Gaze) and compared it to an interface with explanations that appear after clicking (Click), and one with explanations that appear when the user hovers over the recommendation (Hover). In a between-subject study (N=46) we

compared the user experience between these three interfaces. Additionally, we asked the users of Gaze some additional questions. Based on the results, we can conclude that using gaze to dynamically show information does neither increase nor decrease the user experience in terms of *usability, use intention, satisfaction, information sufficiency* and *decision support*. As such, we argue that using gaze in a recommender system is a promising way to balance transparency and information overload without demanding interaction effort of the user.

9. Limitations and future work

Due to the COVID-19 situation, we were forced to conduct the user studies with the Hover and Click online while the user study with Gaze was held in a lab environment. This difference in environment could have caused a bias because of time constraints and social pressure.

Additionally, due to the small number of participants it might be that we did not have enough power to detect differences between the interfaces. Moreover, most of the participants were between 18 and 24 which might have introduced a bias in the direction of more tech-savvy participants than the general population.

For future work, it might be interesting to mix the different interaction techniques which could give users the possibility to trigger extra information in multiple ways.

References

[1] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, M. Graus, Understanding choice overload in recommender systems, in: Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 63–70.

[2] C. He, D. Parra, K. Verbert, Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities, Expert Systems with Applications 56 (2016) 9–27.

[3] J. Kunkel, T. Donkers, L. Michael, C.-M. Barbu, J. Ziegler, Let me explain: Impact of personal and impersonal explanations on trust in recommender systems, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–12.

[4] N. Tintarev, J. Masthoff, A survey of explanations in recommender systems, in: 2007 IEEE 23rd international conference on data engineering workshop, IEEE, 2007, pp. 801–810.

[5] M. Naiseh, N. Jiang, J. Ma, R. Ali, Explainable recommendations in intelligent systems: delivery methods, modalities and risks, in: International Conference on Research Challenges in Information Science, Springer, 2020, pp. 212–228.

[6] M. Naiseh, N. Jiang, J. Ma, R. Ali, Personalising explainable recommendations: Literature and conceptualisation, in: World Conference on Information Systems and Technologies, Springer, 2020, pp. 518–533.

[7] S. Lallé, C. Conati, The role of user differences in customization: a case study in personalization for infovis-based content, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 329–339.

[8] C. Lutteroth, M. Penkar, G. Weber, Gaze vs. mouse: A fast and accurate gaze-only click

alternative, in: Proceedings of the 28th annual ACM symposium on user interface software & technology, 2015, pp. 385–394.

[9] A. Shakil, C. Lutteroth, G. Weber, Codegazer: Making code navigation easy and natural with gaze input, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–12.

[10] F. Jungwirth, M. Murauer, M. Haslgrübler, A. Ferscha, Eyes are different than hands: An analysis of gaze as input modality for industrial man-machine interactions, in: Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference, 2018, pp. 303–310.

[11] A. M. Penkar, C. Lutteroth, G. Weber, Designing for the eye: design parameters for dwell in gaze interaction, in: Proceedings of the 24th Australian Computer-Human Interaction Conference, 2012, pp. 479–488.

[12] K. Grauman, M. Betke, J. Lombardi, J. Gips, G. R. Bradski, Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces, Universal Access in the Information Society 2 (2003) 359–373.

[13] I. S. MacKenzie, An eye on input: research challenges in using the eye for computer input control, in: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, 2010, pp. 11–12.

[14] C. Lankford, Effective eye-gaze input into windows, in: Proceedings of the 2000 symposium on Eye tracking research & applications, 2000, pp. 23–27.

[15] M. Ashmore, A. T. Duchowski, G. Shoemaker, Efficient eye pointing with a fisheye lens, in: Proceedings of Graphics interface 2005, Citeseer, 2005, pp. 203–210.

[16] A. M. Penkar, C. Lutteroth, G. Weber, Eyes only: Navigating hypertext with gaze, in: IFIP Conference on Human-Computer Interaction, Springer, 2013, pp. 153–169.

[17] S. Zhai, C. Morimoto, S. Ihde, Manual and gaze input cascaded (magic) pointing, in: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, 1999, pp. 246–253.

[18] R. Blanch, M. Ortega, Rake cursor: improving pointing performance with concurrent input channels, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2009, pp. 1415–1418.

[19] M. Burnett, Explaining ai: fairly? well?, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, 2020, pp. 1–2.

[20] D. Gunning, Explainable artificial intelligence (xai), Defense Advanced Research Projects Agency (DARPA), nd Web 2 (2017).

[21] A. Holzinger, A. Carrington, H. Müller, Measuring the quality of explanations: the system causability scale (scs), KI-Künstliche Intelligenz (2020) 1–6.

[22] J. L. Herlocker, J. A. Konstan, J. Riedl, Explaining collaborative filtering recommendations, in: Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00, 2000. doi:10.1145/358916.358995.

[23] T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, W.-K. Wong, Too much, too little, or just right? ways explanations impact end users' mental models, in: 2013 IEEE Symposium on Visual Languages and Human Centric Computing, IEEE, 2013, pp. 3–10.

[24] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in eye-tracking protocols, in: Proceedings of the 2000 symposium on Eye tracking research & applications, ACM,

2000, pp. 71–78.

[25] T. Sen, T. Megaw, The effects of task variables and prolonged performance on saccadic eye movement parameters, in: Advances in Psychology, volume 22, Elsevier, 1984, pp. 103–111.

[26] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, C. Newell, Explaining the user experience of recommender systems, User Modeling and User-Adapted Interaction 22 (2012) 441–504.

[27] A. Bangor, P. T. Kortum, J. T. Miller, An empirical evaluation of the system usability scale, Intl. Journal of Human–Computer Interaction 24 (2008) 574–594.