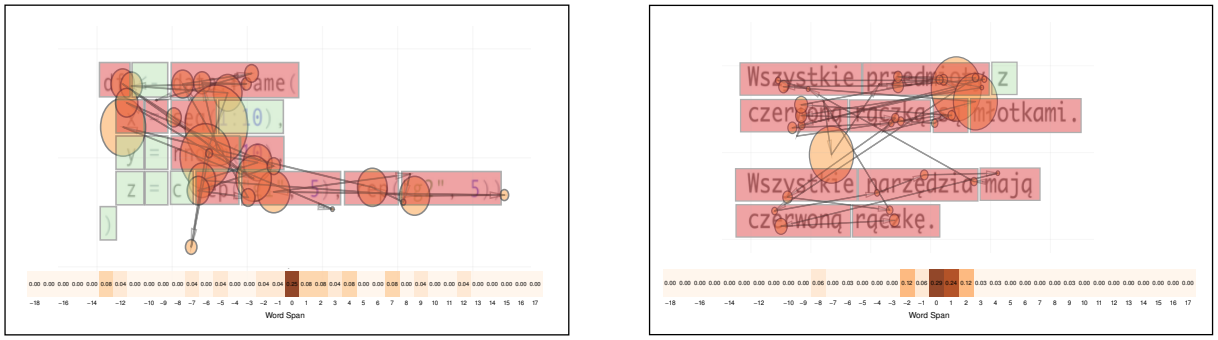
Entropy of Eye Movements While Reading Code or Text

Krzysztof Krejtz*
kkrejtz@swps.edu.pl
kkrejtz@swps.edu.pl
SWPS University
Warsaw, Poland

Andrew T. Duchowski
duchowski@clemson.edu
Clemson University
Clemson, South Carolina, USA
duchowski@clemson.edu

Katarzyna Wisiecka
kwisiecka@swps.edu.pl
SWPS University
Warsaw, Poland
kwisiecka@swps.edu.pl

Izabela Krejtz
ikrejtz@swps.edu.pl
SWPS University
Warsaw, Poland
ikrejtz@swps.edu.pl



(a) Scanpath over code AOIs

(b) Scanpath over textual syllogism AOIs

Figure 1: Exemplary scanpaths of the same participant eye movements over (a) code and (b) textual syllogism. *Note:* Red boxes represent Areas Of Interest (AOIs) hit with at least one fixation. Green boxes depict no fixations detected in the AOI. Circles represent fixations with radius depicts relative fixation duration. Arrows represent saccades. Below the word span probability, used for entropy calculation, is visualized.

ABSTRACT

A new gaze-based analysis method is presented based on word span entropy, suitable for comparison of eye movements collected during reading of code or text. Word span entropy is derived from gaze transition entropy but differs in that the transition matrix represents word span instead of gaze transition between Areas Of Interest (AOIs). Empirical evidence shows that, as expected, an increase in word span entropy is related to shorter response time, especially when reading code, showcasing the metric's analytical utility.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EMIP '22, May 22–27, 2022, Pittsburgh, PA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

eye tracking, gaze entropy, code reading, text reading

ACM Reference Format:
Krzysztof Krejtz, Andrew T. Duchowski, Katarzyna Wisiecka, and Izabela Krejtz. 2022. Entropy of Eye Movements While Reading Code or Text. In *The Tenth International Workshop on Eye Movements in Programming 2022 co-located with 44th International Conference on Software Engineering (ICSE 2022), May 22–27, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The study of eye movements in programming dates back at least to Crosby and Stelovsky [1990], who focused on the impact of expertise on viewing strategies comparing reading of source code with prose. Since then, various eye movement metrics have been used to evaluate gaze over code, most being based on traditional derivations of fixations, fixation durations, pupil size and blink rate,

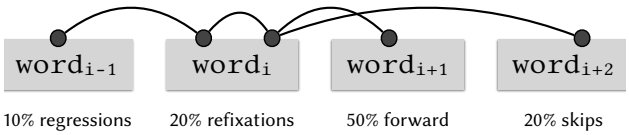


Figure 2: Reading model (from Rayner [1998]).

or scanpath comparison methods such as the Levenshtein string similarity [Sharafi et al. 2015].

Except for the Levenshtein distance, metrics based on fixations and saccades are fairly rudimentary, providing what could be called first-order gaze data analyses. Advanced metrics build on top of these first-order measures to yield second-order interpretation of viewing behavior. Examples include analysis of ambient-focal attention [Krejtz et al. 2016b] and transition entropy analysis [Duchowski 2017].

Transition entropy has previously been developed for gaze comparison [Krejtz et al. 2015] and synthesis [Duchowski et al. 2019], but, to our knowledge, entropy has not yet been applied to analysis of reading behavior. Here, we develop transition entropy from the classical *word span* model of reading [Rayner 1998] as a second-order eye movement metric being able to predict code reading and its comprehension.

2 BACKGROUND

Treating words as boxes over which gaze progresses, reading consists of *re-fixations* of the current word, *skips* forward, or *regressions* backwards. Words are either inspected during a fixation or gaze is moved forward or backward by a saccade [Rayner 1998]. About 10-15% of saccades are regressions to previously fixated words (or lines, when reading multi-line text). Some of these are within-word regressions, considered re-fixations of the word (see Fig. 2). Most reading models are expressed in terms of span distribution, i.e., describing both fixation durations and saccade distributions, e.g., re-fixations, regressions, skips [Thibadeau et al. 1982].

2.1 Reading code

Numerous studies showed differences in reading strategy while reading source code and regular text, e.g., instead of reading code in a linear manner, there is more jumping to circumspect the code, to check function signatures, and to obtain preview of function execution [Feitelson 2019]. The non-linearity of code reading increases with expertise. For example, Busjahn et al. [2015] compared the eye movements of novice and expert programmers in text and Java code reading. Experts read code less linearly than novices and novices read code less linearly than regular text. Using eye tracking, Jbara and Feitelson [2015] demonstrated that, in regular code comprehension, basic repeated patterns tend to attract less attention at the later than initial code segments. In a longitudinal eye tracking study of novice programmers, Al Madi et al. [2021] observed frequency and length effects in reading source code, as programmers progressed through a programming course, analogous to the effects in reading of regular texts.

The non-linearity of code reading, i.e., switching between code elements, may be captured by the analysis of transition entropy. Gaze transition entropy may provide new insights into understanding of code reading in a non-linear manner.

2.2 Gaze transition entropy

Krejtz et al. [2014] introduced a two-stage method for analysis of individual differences in gaze transitions between AOIs during a free viewing task. In the first stage individual sequences of saccades are modeled as first-order Markov chains, wherein the stochastic transition between states depends only on the previous state. In the second stage, Shannon's entropy is computed to allow comparison of gaze transitions between individuals or between experimental conditions. Transition entropy was successfully used for understanding art perception [Krejtz et al. 2015], multimedia learning [Krejtz et al. 2016a], and driver attention [Ebeid and Gwizdka 2018].

Transition matrix entropy was originally developed as a means of expressing gaze transition between AOIs with a 2D matrix, where the matrix ordinate (y-axis) represented the source of gaze transitions and the abscissa (x-axis) contained the destination. By its construction, every Transition Matrix was defined as an $n \times n$ square matrix for the n AOIs defined over the given viewing stimulus.

While Transition Matrix construction serves well for a small number of AOIs, or even a coarsely-defined grid superimposed over the viewing surface, it becomes cumbersome for larger n. Stimuli such as regular text or code typically contain long chains of AOIs, where back-and-forth transitions between the AOIs are expected, unlike over facial AOIs comprised of only eyes, nose, and mouth ($n = 4$). Instead, what is needed is a more intuitive definition of transitions between *relative* AOIs, i.e., gaze transitions between successively (linear) neighbouring AOIs. What is more applicable to reading behavior are transition matrices that embody *word span* more so than direct jumps between source and destination AOIs.

We propose *word span entropy* as a novel metric of eye movement dynamics during reading. Based on previous eye tracking studies of source code reading and the theory of mental models [Johnson-Laird 1983] we assume that during code reading non-linear eye movements depict the construction process of a mental representation of code outcome (better comprehension of the code). Thus we hypothesized that the word span entropy of code reading will predict both the accuracy and response time of code comprehension but not of text reading, i.e., time taken to make a decision regarding outcome of the code or logical implication of the sylogism.

3 IMPLEMENTATION

Gaze transition entropy [Krejtz et al. 2015] relies on gaze transition defined as the conditional probability p_{ij} of viewing the j^{th} AOI given previous viewing of the i^{th} AOI, where p_i is the simple or observed probability of viewing the i^{th} AOI. Entropy measures the complexity of the process $\hat{H}_t = -\sum_{i \in \mathcal{S}} \pi_i \sum_{j \in \mathcal{S}} p_{ij} \log_2 p_{ij}$, with $\mathcal{S} = \{1, \ldots, s\}$ the set of AOIs, π_i the stationary probabilities (estimated through Eigen analysis, see Krejtz et al. [2015] for details), and maximum entropy equal to $\log_2 s$ signifying a uniform distribution of transitions for any of the s AOIs. Minimal entropy of 0 describes a fully deterministic transition sequence. The higher the entropy, the more complexity (randomness) there is in the transitions. Stated another way, entropy refers to the "expected surprise" of a given gaze transition. Minimum entropy of 0 suggests

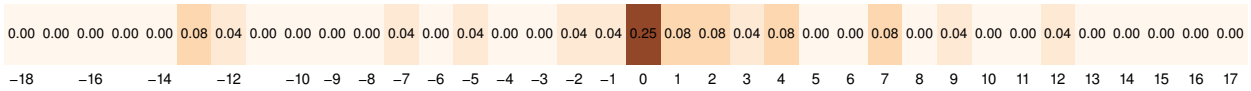


Figure 3: Word span probability used to compute transition entropy. Each cell corresponds to observed probability p_j of word span j. Probability p_0 indicates re-fixation of any given word, p_1 is the probability of skip forward by one word, and p_{-1} is the probability of regressing backward by one word. Remaining word span probabilities are indicated analogously with the length of the word span matrix dependent on the length of the defined sentence. The word span probability matrix shown here corresponds to the scanpath shown in Fig. 1(a).

no expected surprise, meaning that a gaze transition is always expected to the same j^{th} AOI. More formally, the term $-p_{ij} \log_2 p_{ij}$ is the transition's contribution to system entropy, modeled by its probability multiplied by its *surprisal* [Hume and Mailhot 2013].

To facilitate statistical comparison of mean entropies per experimental condition, $\hat{H}_t$ is computed per individual participant and per stimuli and normalized, yielding $H_t = \hat{H}_t / \log_2 s$. This results in a table of entropies (each entropy computed from an individual's transition matrix) for each of the stimuli and each of the participants.

For reading behavior, where an AOI is superimposed atop each word or code chunk, the transition matrix is computed slightly differently as we are mainly concerned with the distance covered by the transition (counted in number of AOIs). The transition matrix, or rather vector in this case, is computed using the probability p_j of transitioning to the given AOI with inter-AOI distance, or word span j. For example, re-fixation of any given word is reflected by probability p_0, a skip forward by one word by p_1, regression backward by one word by p_{-1}, and so on. Word span entropy is computed for these vectors as for transition matrices.

4 METHOD

4.1 Experimental Design and Participants

The study followed a within-subjects experimental design with task (reading code vs. textual syllogism) as the main independent variable. An additional, between-subjects factor controlled in the study was previous experience with R language code (operationalized as completion of a university-level course on statistics with the R language).

Twenty-four university students enrolled in social sciences and humanities bachelor, master, and doctoral programs participated in the study. Results of two were discarded from the analysis due to technical problems during data collection. The final sample consisted of twenty-two participants (17 females) with average age ($M = 27.5, SD = 4.40$). There was no significant difference in age between male and female participants ($t(12.18) = 1.29,\ p = 0.22$). Eight participants (5 females) successfully completed a university course on statistics with the R language, the other 14 were not enrolled in the course. Ten (10) participants, in total, declared they had at least basic knowledge of languages other than R (mostly Python, HTML, and Matlab).

4.2 Procedure and Materials

After signing a consent form, each participant was seated in front of the computer screen. Participants then completed a demographic questionnaire. Next, participants placed their head comfortably on a chin rest (see Fig. 4, top-left). They were introduced to the idea of eye tracking and a five-point calibration and validation began. Each participant attained calibration error below 0.5° visual angle.

The main experimental procedure consisted of two blocks of 5 trials (reading code vs. textual syllogism) given in random order (see Fig. 4). Each trial consisted of three elements: a) fixation point (shown for 500 *ms.* at the location of the first letter in the reading task), b) reading task (R code or textual syllogism, shown for 60 *s.*), and c) a question with three responses to choose from (participants chose by pressing a corresponding key). Between each trial a blank screen was displayed for 1000 *ms.* After each block of trials participants completed the NASA Task Load Index (NASA-TLX) as a reliable and sensitive multidimensional scale of work load [Hart and Staveland 1988].

When reading code, participants were asked to read five (5) lines of R code and to choose an accurate outcome of the code from three alternatives. Code fragments represented common and basic tasks in R usage for statistical computing: data frame creation, data manipulation, graph rendering, ANOVA analysis, and multiple regression statistical analysis.

In text reading, participants read a categorical syllogism consisting of two premises (presented in four lines of text). The participant's task was to decide which conclusion (out of three alternatives) could be drawn from the premises. For example, given two premises: "all fruits in the box are sour" and "all fruits in the box are lemons", the valid conclusion is "all lemons are sour fruits". We decided to use syllogisms for the text reading task because they imply logical reasoning, often used in intelligence batteries [Beauducel and Kersting 2002]. Involving reasoning processes in both tasks, text and code reading presumably makes both experimental conditions equivalent to each other in cognitive requirements. Moreover, both experimental tasks were similar in their visual presentation (same background, font face and size).

For each reading trial, AOIs were defined over words or code chunks as per Busjahn and Tamm [2021] (see Fig. 1 for AOI definition on exemplary text and code stimuli).

4.3 Apparatus & Software

Participants completed all parts of the study procedure in a laboratory room using a Dell Alienware laptop computer with external keyboard, computer mouse, and monitor with 1980×1080 screen resolution (see Fig. 4 top-left). The demographic questionnaire was implemented with Qualtrics XM software. The main procedure was implemented in PsychoPy [Peirce et al. 2019] with the ioHub eye tracker interface for connection to the Gazepoint HD eye tracker.

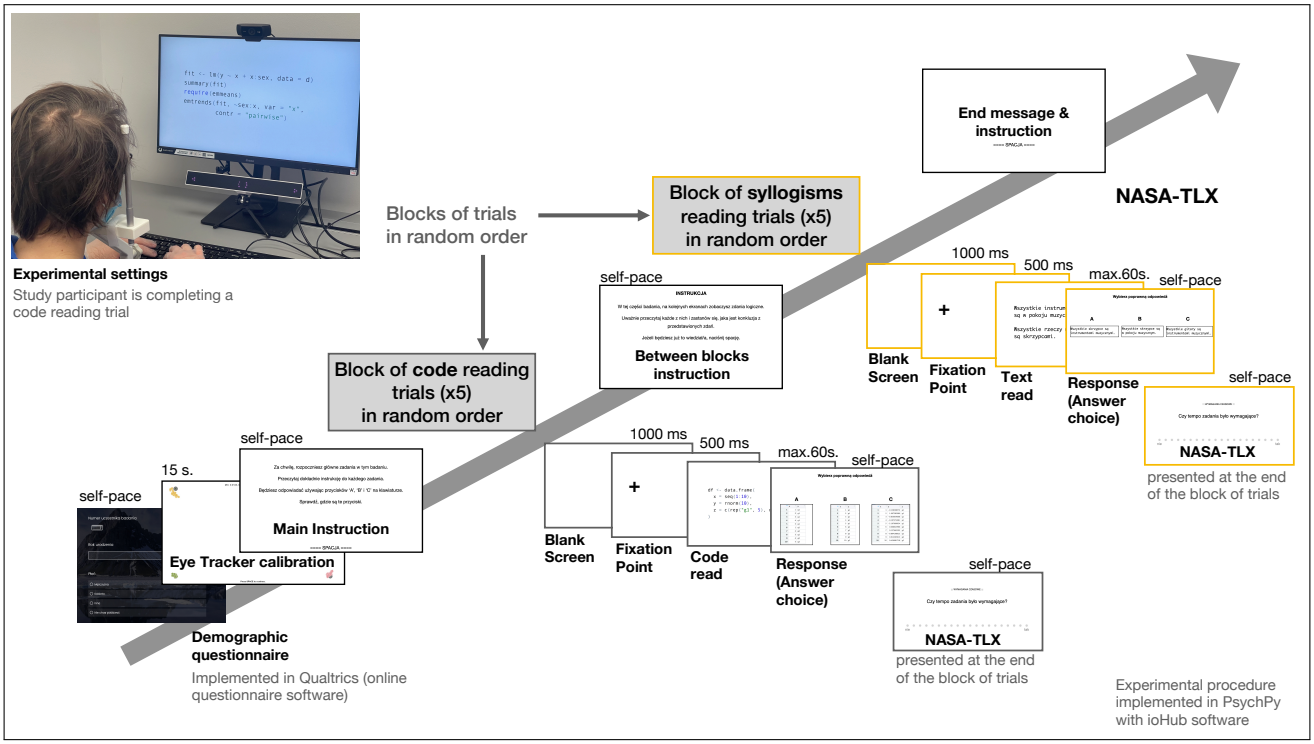


Figure 4: Experimental procedure schematic and experimental settings (top-left inset).

Eye movements were recorded by a GazePoint HD eye tracker with sampling rate of 150 Hz.

4.4 Data Pre-Processing & Analysis

Data processing generally follows the Gaze Analysis Pipeline described by Duchowski [2017], which generally consists of the following steps:

(1) denoise and filter raw gaze data $g_i = (x_i, y_i, t_i)$ to classify raw gaze into fixations $f_i = (x_i, y_i, t_i, d_i)$, where (x_i, y_i) coordinates indicate the position of the gaze point or centroid of the fixation, with t_i indicating the timestamp of the gaze point or fixation and d_i the fixation's duration,
(2) collate fixation-related information for its subsequent statistical comparison,
(3) interpret and visualize statistical tests conducted on processed data.

Visualization of the data at each stage of the pipeline is particularly helpful in fine-tuning parameters.

Statistical analyses were coded in R-computation language (version 4.1.2) [R Core Team 2021] using the lme4 [Bates et al. 2015] and lmerTest [Kuznetsova et al. 2017] packages. The latter package uses Satterthwaite's method to estimate statistical significance for mixed models.

5 RESULTS

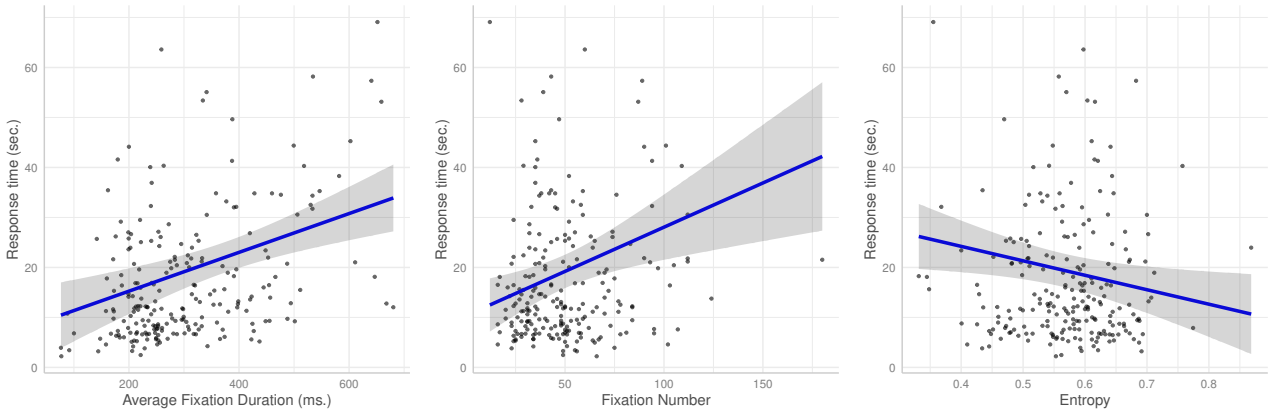
We start by checking whether R language class attendance is related to task accuracy and response time. First, a Chi-squared test of independence was run separately for code and text reading tasks, showing that there was no significant relation between R language class attendance and task accuracy, neither in code ($\chi^2(1) = 1.69$, $p = 0.19$) nor in textual syllogism reading task ($\chi^2(1) = 0.52$, $p = 0.82$).

Next, a two way mixed-design ANOVA with response time as a dependent variable, R language class attendance as a between-subject factor and task as a within-subjects factor revealed a statistically significant main effect of task ($F(1, 20) = 7.93$, $p < 0.02$, $\eta^2 = 0.125$). Response time in code reading task was significantly longer ($M = 20.36s$, $SD = 13.63$) compared to the text reading task ($M = 14.06s$, $SD = 10.73$). Other effects were not significant ($F(1, 20) < 1$).

A similar two-way ANOVA tested differences in task load with NASA-TLX score as a dependent variable. Analysis showed no significant effect of task ($F(1, 19) < 1$), R language class ($F(1, 19) = 2.61$, $p = 0.12$), and interaction ($F(1, 19) < 1$). Since both tasks evoked similar task load and R language class attendance did not differentiate accuracy and response time in either experimental task, we decided not to include these two predictors in the following analyses.

5.1 Predicting response accuracy from eye movement characteristics

Following the hypotheses, to predict the response accuracy we ran a logit Generalized Linear Mixed Model (logit - GLMM) analysis. We included response accuracy as a binomial dependent variable. The fixed effects of task and eye tracking measures (average fixation duration, number of fixations, and entropy), and interaction effects between task type and each eye tracking measure were included in



(a) Average Fixation Duration (b) Number of Fixations (c) Entropy

Figure 5: Main effects for (a) average fixation duration (b) fixation number, and (c) word-span entropy on overall response time. (Note: lines represent the predicted slopes for each main effect; gray areas around each line represent their 95% confidence intervals for the slopes.)

the model. We defined participant nested in task as a random effect. At first the null model was fitted only with the random effect, then the model was updated with task and each eye movement measure predictors as well as interaction terms.

The final model showed a significant main effect of task on response accuracy ($\beta = 4.98$, $z = 1.97$, $p < 0.05$). Response accuracy in text reading was significantly higher ($M = 66\%$, $SD = 47$) than in code reading ($M = 52\%$, $SD = 50$). None of the other model effects was statistically significant.

5.2 Predicting response time from eye movement characteristics

Linear Mixed Model (LMM) analysis was run with response time as dependent variable. The structure of both fixed effects and random effects was the same as for GLMM analysis. Additionally, response accuracy was added as a fixed factor to the model.

The final model revealed a significant main effect of accuracy on response time ($\beta = -3.50$, $t(205.39) = 2.35$, $p < 0.02$). Response time was significantly lower when the response was accurate ($M = 15.30s$, $SD = 11.37$) than when it was not ($M = 19.96s$, $SD = 13.88$).

We also noted main effects of average fixation duration ($\beta = 0.04$, $t(106.26) = 3.94$, $p < 0.001$, see Fig. 5a) and number of fixations ($\beta = 0.18$, $t(132.51) = 3.15$, $p < 0.01$, see Fig. 5b). Increase in average fixation duration and number of fixations predicts longer response time. A main effect of entropy was also significant ($\beta = -28.92$, $t(205.37) = 2.39$, $p < 0.02$, see Fig. 5c). In line with predictions, increased *word span* entropy suggests shorter response time.

The main effects of number of fixations and entropy were moderated by task. Interaction of task and number of fixations was significant ($\beta = -0.19$, $t(170.51) = 2.63$, $p < 0.01$, see Fig. 6a). Simple slope analysis showed that the slope of the relation between number of fixations and response time differs significantly between text and code reading ($t(189) = 2.52$, $p = 0.01$). Increase in number of fixations in code reading significantly predicts an increase in response time ($\beta = 0.18$, $SE = 0.06$) but it does not predict response time in text reading ($\beta = -0.01$, $SE = 0.04$), see Fig. 6a.

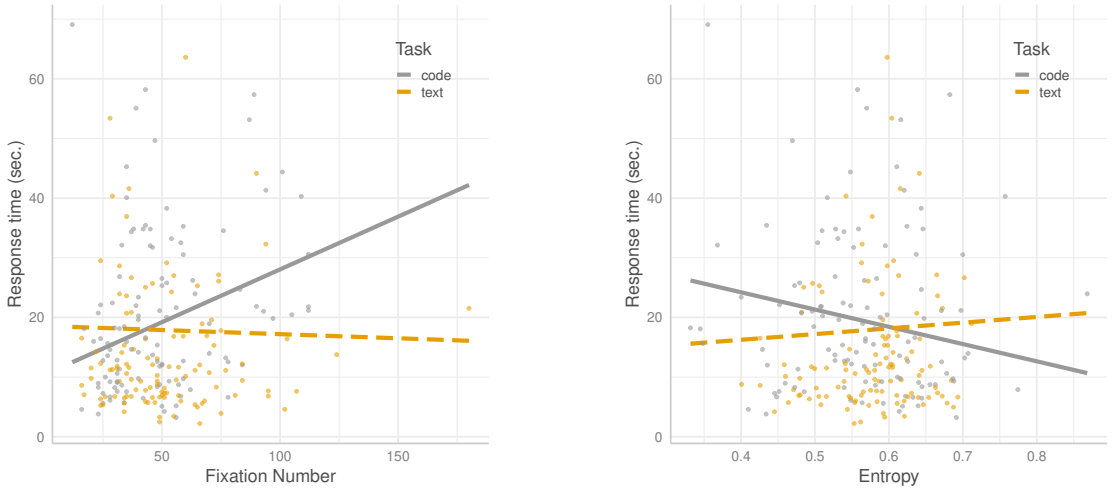
The interaction of task and entropy was marginally significant ($\beta = 38.49$, $t(219.50) = 1.82$, $p = 0.069$, see Fig. 6b). Simple slope analysis showed that the difference in slopes between text and code reading is marginally significant ($t(228) = 1.77$, $p = 0.077$). The slope of relation between entropy as a predictor and response time is negative for code reading ($\beta = -28.92$, $SE = 12.40$) but for text reading it is slightly positive ($\beta = 9.57$, $SE = 17.80$), see Fig. 6b.

6 DISCUSSION & CONCLUSIONS

The present study examined the utility of a novel eye movement measure, word span entropy, in predicting reading and comprehension of source code and text. The two reading tasks involved reasoning, evoking similar work load. Response time was predicted by average fixation duration, number of fixations, and word span entropy. Longer fixation duration indicated more time needed to respond. As predicted, the higher the entropy the shorter the response time when reading code but not textual syllogisms.

To understand the role of word span entropy in predicting response time, an intuitive interpretation of entropy is helpful. The higher the entropy, the more complex (random) the visual switching between different code chunks, since transition from source AOI to any destination AOI (code chunk) is equally likely. Higher word span entropy indicates a more random scanpath (i.e., not following expected reading order from word to word) over code, leading to faster response time linked to code comprehension. This result is in line with several studies showing that code reading is substantially different from text reading by involving more eye jumps in various directions (even between code lines) [Busjahn et al. 2015; Feitelson 2019; McChesney and Bond 2019].

One limitation of word span entropy is that it is dependent on the number and size of AOIs defined. Non-uniformly sized AOIs



(a) No. of Fixations and Task Type (block)

(b) Entropy and Task Type (block)

Figure 6: Interaction effects of (a) number of fixations and task type and (b) entropy and task type on response time. *Note*: lines represent the predicted slopes in each task (code vs. text reading).

are likely to garner greater probability of fixation on larger AOIs. Ideally, AOI sizes and their number should be held constant between conditions. Exact effects of AOI non-uniformity on word span entropy need further investigation.

To conclude, word span entropy is a promising measure of the reading process that is simple to implement. As a second-order metric, it complements other eye movement measures useful in interpreting programming behavior while reading and comprehending source code. The full understanding of relationship between gaze entropy of code reading needs further experimental research. Future studies may examine differences in word span entropy between experts and novices while reading code.

REFERENCES

Naser Al Madi, Cole S. Peterson, Bonita Sharif, and Jonathan I. Maletic. 2021. From Novice to Expert: Analysis of Token Level Effects in a Longitudinal Eye Tracking Study. In *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*. 172–183. https://doi.org/10.1109/ICPC52881.2021.00025

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* 67, 1 (2015), 1–48. https://doi.org/10.18637/jss.v067.i01

André Beauducel and M Kersting. 2002. Fluid and cristallized intelligence and the Berlin Model of Intelligence Structure (BIS). *European Journal of Psychological Assessment* 18 (01 2002), 97–112.

Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H. Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye Movements in Code Reading: Relaxing the Linear Order. In *2015 IEEE 23rd International Conference on Program Comprehension*. 255–265. https://doi.org/10.1109/ICPC.2015.36

Teresa Busjahn and Sascha Tamm. 2021. A Deeper Analysis of AOI Coverage in Code Reading. In *ACM Symposium on Eye Tracking Research and Applications*. ACM, Virtual Event Germany, 1–7. https://doi.org/10.1145/3448018.3457422

Martha E. Crosby and Jan Stelovsky. 1990. How do we read algorithms? A case study. *Computer* 23, 1 (1990), 25–35. https://doi.org/10.1109/2.48797

Andrew T. Duchowski. 2017. *Eye Tracking Methodology: Theory & Practice* (3rd ed.). Springer-Verlag, Inc., London, UK.

Andrew T. Duchowski, Sophie Jörg, Jaret Screws, Nina A. Gehrer, Michael Schönenberg, and Krzysztof Krejtz. 2019. Guiding Gaze: Expressive Models of Reading and Face Scanning. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, CO) *(ETRA '19)*. Association for Computing Machinery, New York, NY, Article 25, 9 pages. https://doi.org/10.1145/3314111.3319848

Islam Akef Ebeid and Jacek Gwizdka. 2018. Real-time gaze transition entropy. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, Warsaw Poland, 1–3. https://doi.org/10.1145/3204493.3208340

Dror G. Feitelson. 2019. Eye Tracking and Program Comprehension. In *2019 IEEE/ACM 6th International Workshop on Eye Movements in Programming (EMIP)*. 1–1. https://doi.org/10.1109/EMIP.2019.00008

Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Elizabeth Hume and Frédéric Mailhot. 2013. The Role of Entropy and Surprisal in Phonologization and Language Change. In *Origins of Sound Change: Approaches to Phonologization*, Alan C. L. Yu (Ed.). Oxford University Press, Oxford, UK, 29–47.

Ahmad Jbara and Dror G. Feitelson. 2015. How Programmers Read Regular Code: A Controlled Experiment Using Eye Tracking. In *2015 IEEE 23rd International Conference on Program Comprehension*. 244–254. https://doi.org/10.1109/ICPC.2015.35

Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press.

Krzysztof Krejtz, Andrew Duchowski, Izabela Krejtz, Agnieszka Szarkowska, and Agata Kopacz. 2016b. Discerning Ambient/Focal Attention with Coefficient K. *ACM Trans. Appl. Percept.* 13, 3, Article 11 (may 2016), 20 pages. https://doi.org/10.1145/2896452

Krzysztof Krejtz, Andrew Duchowski, Tomasz Szmidt, Izabela Krejtz, Fernando González Perilli, Ana Pires, Anna Vilaro, and Natalia Villalobos. 2015. Gaze Transition Entropy. *ACM Transactions on Applied Perception* 13, 1, Article 4 (Dec. 2015), 20 pages. https://doi.org/10.1145/2834121

Krzysztof Krejtz, Andrew T. Duchowski, Izabela Krejtz, Agata Kopacz, and Piotr Chrząstowski-Wachtel. 2016a. Gaze transitions when learning with multimedia. *Journal of Eye Movement Research* 9, 1 (2016). https://doi.org/10.16910/jemr.9.1.5

Krzysztof Krejtz, Tomasz Szmidt, Andrew T. Duchowski, and Izabela Krejtz. 2014. Entropy-Based Statistical Analysis of Eye Movement Transitions. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Safety Harbor, Florida) *(ETRA '14)*. Association for Computing Machinery, New York, NY, USA, 159–166. https://doi.org/10.1145/2578153.2578176

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software* 82, 13 (2017), 1–26. https://doi.org/10.18637/jss.v082.i13

Ian McChesney and Raymond Bond. 2019. Eye tracking analysis of computer program comprehension in programmers with dyslexia. *Empirical Software Engineering* 24, 3 (June 2019), 1109–1154. https://doi.org/10.1007/s10664-018-9649-y

Jonathan Peirce, Jeremy R. Gray, Sol Simpson, Michael MacAskill, Richard Höchenberger, Hiroyuki Sogo, Erik Kastman, and Jonas Kristoffer Lindeløv. 2019. PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods* 51, 1 (2019), 195–203. https://doi.org/10.3758/s13428-018-01193-y

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Keith Rayner. 1998. Eye Movements in Reading and Information Processing: 20 Years of Research. *Psychological Bulletin* 124, 3 (1998), 372–422.

Zohreh Sharafi, Timothy Shaffer, Bonita Sharif, and Yann-Gaël Guéhéneuc. 2015. Eye-Tracking Metrics in Software Engineering. In *2015 Asia-Pacific Software Engineering Conference (APSEC)*. 96–103. https://doi.org/10.1109/APSEC.2015.53

Robert Thibadeau, Marcel Adam Just, and Patricia A. Carpenter. 1982. A Model of the Time Course and Content of Reading. *Cognitive Science* 6 (1982), 157–203.