Interactive Assessment Tool for Gaze-based Machine Learning Models in Information Retrieval

Pablo Valdunciel
pablo.valdunciel@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Omair Shahzad Bhatti
ombh01@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Michael Barz
michael.barz@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Oldenburg University
Oldenburg, Germany

Daniel Sonntag
daniel.sonntag@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Oldenburg University
Oldenburg, Germany

ABSTRACT

Eye movements were shown to be an effective source of implicit relevance feedback in information retrieval tasks. They can be used to, e.g., estimate the relevance of read documents and expand search queries using machine learning. In this paper, we present the Reading Model Assessment tool (ReMA), an interactive tool for assessing gaze-based relevance estimation models. Our tool allows experimenters to easily browse recorded trials, compare the model output to a ground truth, and visualize gaze-based features at the token- and paragraph-level that serve as model input. Our goal is to facilitate the understanding of the relation between eye movements and the human relevance estimation process, to understand the strengths and weaknesses of a model at hand, and, eventually, to enable researchers to build more effective models.

CCS CONCEPTS

• **Human-centered computing → Visualization systems and tools**; • **Information systems → Users and interactive retrieval**.

KEYWORDS

eye tracking, reading, data visualization, relevance estimation, information retrieval, interactive model assessment

ACM Reference Format:
Pablo Valdunciel, Omair Shahzad Bhatti, Michael Barz, and Daniel Sonntag. 2022. Interactive Assessment Tool for Gaze-based Machine Learning Models in Information Retrieval. In *ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '22), March 14–18, 2022, Regensburg, Germany.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3498366.3505834

1 INTRODUCTION

Research from the past decade has revealed the potential of eye tracking to improve interactive information retrieval (IIR) systems. For instance, eye tracking has enabled the development of methods for implicit query expansion [9], query reformulation [14], query prediction [1, 13], and relevance estimation [6, 15, 29]. Typically, these methods rely on machine learning models with encoded eye movements as input. Common features are based on eye events over the raw gaze signal (timestamped 2D coordinates) like fixations and saccades. Fixations are phases in which the eye remains still while saccades are rapid motions between two fixations [19] (pp. 13-15). The sequence of all eye events of a user is denoted as scanpath. In this work, we present the **Re**ading **M**odel **A**ssessment tool (ReMA), an interactive tool for analyzing scanpaths from relevance judgement tasks and for assessing gaze-based relevance estimation models, which can be used to build interactive information retrieval (IIR) systems.

CHIIR '22, March 14–18, 2022, Regensburg, Germany

ACM ISBN 978-1-4503-9186-3/22/03.
https://doi.org/10.1145/3498366.3505834

2 RELATED WORK

Our work is related to other tools in the context of reading behaviour analysis. For instance, GazePlot enables reading performance analysis for children [32]. EyeMap allows researchers to analyze fixations and saccades at the word level [31]. Both offer a scanpath visualization as text overlays: eye tracking data is presented as a gaze plot in which fixations are depicted as circles and saccades as lines. However, "actual scanpath records are usually quite complex, and can be difficult to interpret and compare" [16]. Other tools implemented more intuitive visualizations that for example aggregate the gaze data at the word level by mapping the gaze data to objects of the Document Object Model (DOM) of a web page [5, 18, 28]. WebEyeMapper and WebLogger [28] and WebGazeAnalyzer [5] introduced this approach. Hienert et al. [18] use a similar mapping approach in the Reading Protocol tool. It allows experimenters to more effectively analyze eye movements on arbitrary areas of interest. They use this gaze-to-object mapping to generate a heat map that visualizes the summed fixation durations at the word level. Davari et al. [13] use this tool to investigate the role of word fixations in query term prediction. Buscher et al. [8] introduced the concept of attentive documents that keep track of a user's perceived relevance based on its eye movements. Their system may highlight text passages that were previously read and not skimmed. Eyekit is a recent Python package that supports the analysis of reading behaviour [10]. It provides different visualization

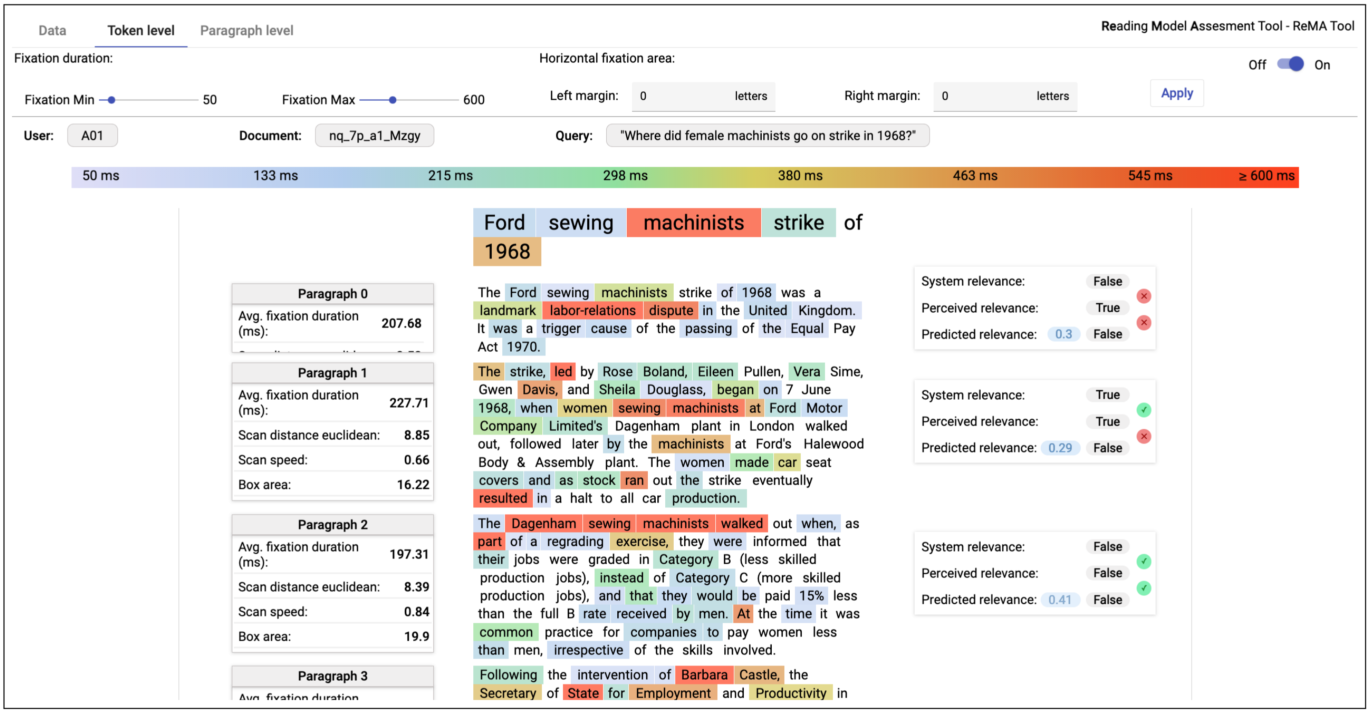

Figure 1: Screenshot of the interface of the ReMA tool, our interactive reading model assessment tool. It shows a Wikipedia article with the token-level heat map and paragraph-level features, and the relevance information per paragraph.

techniques like gaze plots and character-based heat maps, which work similar to the word-based heat map from Hienert et al. [18]. Moreover, it offers gaze-based features extraction from scanpaths that originate from a reading activity.

3 EXAMPLE DATASET

To demonstrate the functionality of the ReMA tool, we use data from our gazeRE dataset ($n = 24$), which is available on Github[1]. The participants were asked to read documents of different lengths and to judge, per paragraph, whether it provided an answer to a previously shown trigger question. The goal was to collect eye movement data during relevance judgement tasks. We used this data to model the relation between the recorded eye movement data and the perceived relevance using machine learning. The stimuli from this study are pairs of trigger questions and documents from two corpora: We used a subset of 12 documents from the g-REL corpus which includes short news articles that fit on one page [6, 17] and 12 pairs from the Google Natural Questions (NQ) corpus, which includes multi-paragraph documents from Wikipedia that require the user to scroll down to read the whole text [21]. Both corpora include binary relevance annotations per paragraph (system relevance). Also, participants rated a paragraph as relevant or non-relevant (perceived relevance) which is the classification target. Besides, we stored the exact token positions for each document and for each point in time. With tokens, we refer to text segments that are divided by space characters, i.e., mostly words with or without subsequent punctuation. In this work, we use a binary classification model trained on this dataset. The model is based on the scikit-learn machine learning framework [24] and includes three components: At training time, we apply the oversampling technique SMOTE [11] from the imbalanced-learn package [22] because relevant paragraphs are underrepresented. Further, we apply the standard feature scaler which removes the mean and scales features to unit variance. The last component in this pipeline is the actual classifier, a support vector classifiers with default parameters (`kernel = "rbf", C = 1`). A detailed description of the user study, the relevance models, and their evaluation can be found in Barz et al. [2].

[1] The gazeRE dataset can be downloaded from https://github.com/DFKI-Interactive-Machine-Learning/gazeRE-dataset

4 READING MODEL ASSESSMENT TOOL

We implement the **Re**ading **M**odel **A**ssessment tool (ReMA) for interactively assessing relevance-judgement models that take a scanpath as input to predict the perceived relevance. It allows researchers to easily browse recorded trials, which include a text-based stimulus, a user's scanpath, and the system relevance and perceived relevance. As it is rarely helpful to visualize the fixation and saccade sequences to understand and compare scanpaths [16], we show the text-based stimulus along with paragraph-level features and a token-level heat map. In addition, it shows the predicted relevance estimate and highlights whether it agrees or disagrees with the perceived relevance (ground truth) which allows the researcher to more efficiently assess the model and to understand its strengths and weaknesses.

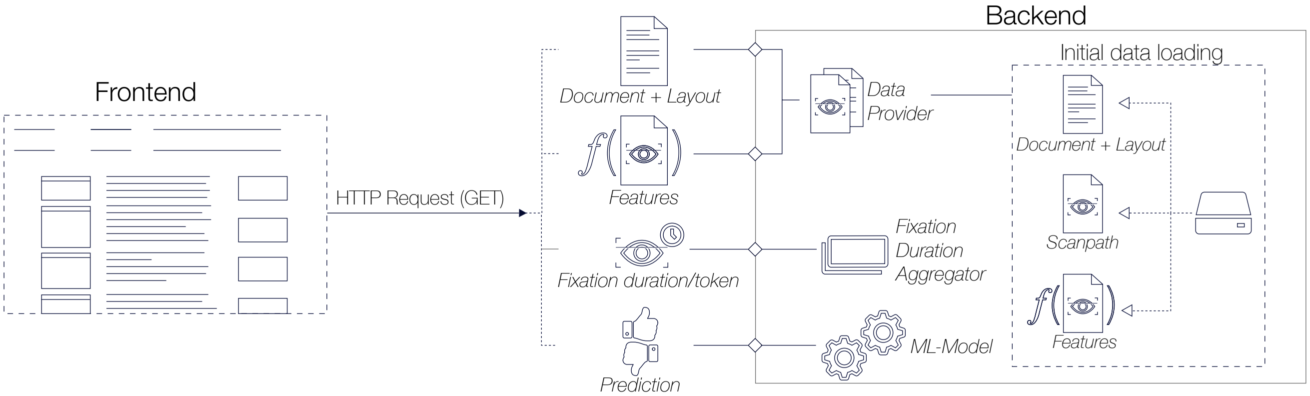

Figure 2: Architectural overview of our interactive model assessment tool. The backend serves all information to the frontend including the stimuli, the extracted features, and the model predictions.

The user interface shows a single trial with the text-based stimulus as its central element (see Figure 1). A trial can be selected via the *Data* tab by providing a participant's acronym and a document ID. The corresponding query is shown below the tab area. The *Token level* and *Paragraph level* tabs can be used to configure the visualization which includes the token-level heat map, the paragraph-level feature display, and the relevance ratings. The size and position of each token corresponds to the original layout from the user study. However, we scale the layout depending on the resolution of the connected display. The web-based frontend is built using AngularJS, HTML, CSS and Typescript. It is responsive and can adapt its layout to different screen sizes. All data can be queried from a single backend that connects the example dataset (see Figure 2). It offers functions for loading a document, the token-level heat map, and extracted features per paragraph. Also, it integrates the pre-trained machine learning model to provide relevance estimates for a selected trial. The backend REST API is implemented using the Python framework Flask. A demo of our tool can be found online[2].

4.1 Token-level Heat Map

We integrate the token-level heat map, which encodes a scanpath by accumulating fixation durations per token, as proposed in Hienert et al. [18]. We extend the heat map generation by a configurable perceptual span setting which allows experimenters to more accurately estimate the information that was actually processed by a reader. The heat map is generated by coloring the background of each token based on the duration of all fixations (sum) that hit the token area. The range of considered fixation durations can be set using two sliders in the *Token level* tab as shown in Figure 1. The resulting color legend is shown above the stimulus and replicates the coloring in Hienert et al. [18]. Tokens that were fixated shorter than the minimum threshold are not colored. Fixation durations longer than the maximum threshold result in a red background color. However, the region from which useful information is acquired during a fixation, also known as the perceptual span, is wider than a single character [27]. Research using the classical paradigm of the gaze-contingent moving window has shown that in English and other alphabetic languages read from left to right, the perceptual span extends 3-4 letters to the left, and up to 14-15 letters to the right for a given fixation point [23]. But, the extent of the perceptual span is not constant: its size is influenced by linguistic parameters such as the readability of the text [25], the frequency of words [26], and the linguistic ability of the reader [12]. We extend the token-level heat map to account for the perceptual span (see Figure 1): Our tool allows researchers to configure a perceptual span by setting a left margin m_l and a right margin m_r, which define how many letters to the left and right of fixation are considered for accumulating the fixation durations per token. The duration of the current fixation is added for all tokens, which overlap with this region. Hereby, one letter translates to a fixed number of pixels based on the font size.

4.2 Paragraph-level Features

Per paragraph, we display a box with features extracted from the longest partial scanpath for this paragraph to its left. To encode the eye movements of a user for a certain paragraph, we have to extract coherent gaze sequences that lie within the paragraph area. We refer to these partial scanpaths as visits. A user might visit a paragraph multiple times during the relevance judgement process. We extract all visits to a paragraph with a minimum length while ignoring short gaps: As long as there is a pair of two subsequent visits with a gap shorter than 0.2 s, these visits are merged. All visits that satisfy a minimum length of 3 s are kept. A common way to encode a scanpath in a meaningful way is to extract handcrafted features like the `scan_hv_ratio`: the horizontal to vertical ratio of saccade amplitudes [19] (p. 442). We extract a set of 17 features[3], including the `scan_hv_ratio`, which have been used to model the perceived relevance of short news articles in Bhattacharya et al. [6] and Barz et al. [2]. Four features are based on fixation events, eight are based on saccadic movements, and five are based on the area spanned by all fixations. In the *Paragraph level* tab, researchers can select features that shall be displayed.

[2]A prototype of the Reading Model Assessment Tool, ReMA, can be accessed via https://iml.dfki.de/demos/rematool/.

[3]The source code for feature extraction is available on https://github.com/DFKI-Interactive-Machine-Learning/gazeRE-dataset/tree/main/features

4.3 Relevance Model Assessment

For each paragraph, our ReMA tool shows the system relevance, the perceived relevance collected during the user study, and the predicted relevance using the pre-trained model. The relevance values can be either *True* or *False*, indicating that this paragraph is deemed to be relevant to the corresponding query or not. We display the model's certainty in terms of its probability estimate $p \in [0, 1]$ for the *True* class (relevant): 1 indicates that the model is certain that the trial includes a relevant paragraph, 0 indicates certainty for a non-relevant instance. The certainty is shown in blue with an opacity proportional to its value, i.e. the closer p is to 0 or 1, the more opaque the value is shown. In addition, we show circular badges indicating whether the participant's perceived relevance agrees with the system relevance and whether the model correctly predicted the perceived relevance. This enables a more efficient assessment of the model performance in context.

5 DISCUSSION

Our interactive Reading Model Assessment tool (ReMA) renders the output of a relevance estimation model along with different visualizations of the input. We believe that this kind of user interface can help researchers to better understand the behavior of a machine learning model and to identify the strengths and weaknesses of such models. We implement three views for this purpose, the token-level heat map that aggregates and visualizes fixation times per token using a color scale, the paragraph-level display, and the relevance comparison. Future investigations should aim to confirm these potential benefits. We envision an interactive machine learning cycle that integrates concepts from explainable artificial intelligence to add more transparency in the modelling process and, by that, allows domain experts to effectively and incrementally improve the resulting models [20]. It is important to identify useful visualizations from the multitude of available techniques, see Blascheck et al. [7] for an overview. Another aspect that should be considered in future work is to incorporate the gaze-estimation error in the data analysis and modelling process [3]. Furthermore, the extension to other domains such as visual search [4, 30] should be considered.

6 CONCLUSION

We implemented an interactive tool for assessing gaze-based relevance estimation models. The parallel visualization of a textual stimulus, extracted features, and a relevance estimate along with the ground truth increases the model transparency and should allow eye tracking researchers to understand the strengths and weaknesses or design issues of the model at hand.

ACKNOWLEDGMENTS

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01JD1811C (GeAR) and in the Software Campus project SciBot.

REFERENCES

[1] Antti Ajanki, David R. Hardoon, Samuel Kaski, Kai Puolamäki, and John Shawe-Taylor. 2009. Can eyes reveal interest? Implicit queries from gaze patterns. *User Modeling and User-Adapted Interaction* 19, 4 (Oct. 2009), 307–339. https://doi.org/10.1007/s11257-009-9066-4

[2] Michael Barz, Omair Shahzad Bhatti, and Daniel Sonntag. 2021. Implicit Estimation of Paragraph Relevance from Eye Movements. *Frontiers in Computer Science* (2021). https://doi.org/10.3389/fcomp.2021.808507

[3] Michael Barz, Florian Daiber, Daniel Sonntag, and Andreas Bulling. 2018. Error-aware gaze-based interfaces for robust mobile gaze interaction. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*, Bonita Sharif and Krzysztof Krejtz (Eds.). ACM, 24:1–24:10. https://doi.org/10.1145/3204493.3204536

[4] Michael Barz, Sven Stauden, and Daniel Sonntag. 2020. Visual Search Target Inference in Natural Interaction Settings with Machine Learning. In *ACM Symposium on Eye Tracking Research and Applications (ETRA '20 Full Papers)*, Andreas Bulling, Anke Huckauf, Eakta Jain, Ralph Radach, and Daniel Weiskopf (Eds.). Association for Computing Machinery, 1–8. https://doi.org/10.1145/3379155.3391314

[5] David Beymer and Daniel M. Russell. 2005. WebGazeAnalyzer: a system for capturing and analyzing web reading behavior using eye gaze. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. ACM, Portland OR USA, 1913–1916. https://doi.org/10.1145/1056808.1057055

[6] Nilavra Bhattacharya, Somnath Rakshit, and Jacek Gwizdka. 2020. Towards Real-time Webpage Relevance Prediction UsingConvex Hull Based Eye-tracking Features. In *Symposium on Eye Tracking Research and Applications*. ACM, Stuttgart Germany, 1–10. https://doi.org/10.1145/3379157.3391302

[7] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum* 36, 8 (Dec. 2017), 260–284. https://doi.org/10.1111/cgf.13079 Publisher: John Wiley & Sons, Ltd.

[8] Georg Buscher, Andreas Dengel, Ralf Biedert, and Ludger V. Elst. 2012. Attentive documents: Eye tracking as implicit feedback for information retrieval and beyond. *ACM Transactions on Interactive Intelligent Systems* 1, 2 (Jan. 2012), 1–30. https://doi.org/10.1145/2070719.2070722

[9] Georg Buscher, Andreas Dengel, Ludger van Elst, and Florian Mittag. 2008. Generating and using gaze-based document annotations. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. ACM, Florence Italy, 3045–3050. https://doi.org/10.1145/1358628.1358805

[10] Jon W. Carr, Valentina N. Pescuma, Michele Furlan, Maria Ktori, and Davide Crepaldi. 2021. Algorithms for the automated correction of vertical drift in eye-tracking data. *Behavior Research Methods* (June 2021). https://doi.org/10.3758/s13428-021-01554-0

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357. https://doi.org/10.1613/jair.953

[12] Wonil Choi, Matthew W. Lowder, Fernanda Ferreira, and John M. Henderson. 2015. Individual differences in the perceptual span during reading: Evidence from the moving window technique. *Attention, Perception, & Psychophysics* 77, 7 (Oct. 2015), 2463–2475. https://doi.org/10.3758/s13414-015-0942-1

[13] Masoud Davari, Daniel Hienert, Dagmar Kern, and Stefan Dietze. 2020. The Role of Word-Eye-Fixations for Query Term Prediction. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. ACM, Vancouver BC Canada, 422–426. https://doi.org/10.1145/3343413.3378010

[14] Carsten Eickhoff, Sebastian Dungs, and Vu Tran. 2015. An Eye-Tracking Study of Query Reformulation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Santiago Chile, 13–22. https://doi.org/10.1145/2766462.2767703

[15] Anna Maria Feit, Lukas Vordemann, Seonwook Park, Caterina Berube, and Otmar Hilliges. 2020. Detecting Relevance during Decision-Making from Eye Movements for UI Adaptation. In *ACM Symposium on Eye Tracking Research and Applications*. ACM, Stuttgart Germany, 1–11. https://doi.org/10.1145/3379155.3391321

[16] Joseph H. Goldberg and Jonathan I. Helfman. 2010. Visual Scanpath Representation. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 203–210. https://doi.org/10.1145/1743666.1743717

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

[18] Daniel Hienert, Dagmar Kern, Matthew Mitsui, Chirag Shah, and Nicholas J. Belkin. 2019. Reading Protocol: Understanding what has been Read in Interactive Information Retrieval Tasks. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. ACM, Glasgow Scotland UK, 73–81. https://doi.org/10.1145/3295750.3298921

[19] Kenneth Holmqvist and Richard Andersson. 2011. *Eye tracking: A comprehensive guide to methods, paradigms and measures*. Lund Eye-Tracking Research Institute, Lund, Sweden.

[20] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (Atlanta, Georgia, USA) *(IUI '15)*. Association for Computing Machinery, New York, NY, USA, 126–137. https://doi.org/10.1145/2678025.2701399

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (Aug. 2019), 453–466. https://doi.org/10.1162/tacl_a_00276

[22] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365.html

[23] George W. McConkie and Keith Rayner. 1976. Asymmetry of the perceptual span in reading. *Bulletin of the Psychonomic Society* 8, 5 (Nov. 1976), 365–368. https://doi.org/10.3758/BF03335168

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

[25] Keith Rayner. 1986. Eye movements and the perceptual span in beginning and skilled readers. *Journal of Experimental Child Psychology* 41, 2 (1986), 211–236. https://doi.org/10.1016/0022-0965(86)90037-8 Place: Netherlands Publisher: Elsevier Science.

[26] Keith Rayner, Simon P. Liversedge, Sarah J. White, and Dorine Vergilino-Perez. 2003. Reading disappearing text: Cognitive control of eye movements. *Psychological Science* 14, 4 (2003), 385–388. https://doi.org/10.1111/1467-9280.24483 Place: United Kingdom Publisher: Blackwell Publishing.

[27] Keith Rayner, Timothy J. Slattery, and Nathalie N. Bélanger. 2010. Eye movements, the perceptual span, and reading speed. *Psychonomic Bulletin & Review* 17, 6 (Dec. 2010), 834–839. https://doi.org/10.3758/PBR.17.6.834

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

[29] Jarkko Salojärvi, Kai Puolamäki, Jaana Simola, Lauri Kovanen, Ilpo Kojo, and Samuel Kaski. 2005. Inferring Relevance from Eye Movements: Feature Extraction. *Publications in Computer and Information Science* Report A82 (2005), 45–67. http://research.ics.aalto.fi/events/eyechallenge2005/irem-2005-03-03.pdf Publisher: Helsinki University of Technology.

[30] Sven Stauden, Michael Barz, and Daniel Sonntag. 2018. Visual Search Target Inference Using Bag of Deep Visual Words. In *KI 2018: Advances in Artificial Intelligence - 41st German Conference on AI, Berlin, Germany, September 24-28, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 11117)*, Frank Trollmann and Anni-Yasmin Turhan (Eds.). Springer, 297–304. https://doi.org/10.1007/978-3-030-00111-7_25

[31] Siliang Tang, Ronan G. Reilly, and Christian Vorstius. 2012. EyeMap: a software system for visualizing and analyzing eye movement data in reading. *Behavior Research Methods* 44, 2 (June 2012), 420–438. https://doi.org/10.3758/s13428-011-0156-y

[32] Oleg Špakov, Harri Siirtola, Howell Istance, and Kari-Jouko Räihä. 2017. Visualizing the Reading Activity of People Learning to Read. *Journal of Eye Movement Research* 10, 5 (Nov. 2017). https://doi.org/10.16910/jemr.10.5.5