Team ReadMe at CMCL 2021 Shared Task: Predicting Human Reading Patterns by Traditional Oculomotor Control Models and Machine Learning

Alişan Balkoca, A. Can Algan, Cengiz Acartürk
Cognitive Science Department, Middle East Technical University
Çağrı Çöltekin
Department of Linguistics, University of Tübingen

Abstract

This system description paper describes our participation in CMCL 2021 shared task on predicting human reading patterns. Our focus in this study is making use of well-known, traditional oculomotor control models and machine learning systems. We present experiments with a traditional oculomotor control model (the EZ Reader) and two machine learning models (a linear regression model and a recurrent network model), as well as combining the two different models. In all experiments we test effects of features well-known in the literature for predicting reading patterns, such as frequency, word length and word predictability. Our experiments support the earlier findings that such features are useful when combined. Furthermore, we show that although machine learning models perform better in comparison to traditional models, combination of both gives a consistent improvement for predicting multiple eye tracking variables during reading.

1 Introduction

Oculomotor control in reading has been a well-established domain in eye tracking research. From the perspective of perceptual and cognitive mechanisms that drive eye movement control, the characteristics of the visual stimuli is better controlled in reading research than visual scene stimuli. Several computational models have been developed for the past two decades, which aimed at modeling the relationship between a set of independent variables, such as word characteristics and dependent variables, such as fixation duration and location (Kliegl et al., 2006).

Based on the theoretical and experimental research in reading, the leading independent variables include the frequency of a word in daily use, the length of the word and its sentential predictability. The term *sentential predictability* (or word predictability) is used to define predictability score which is the probability of guessing a word from the sequence of previous words of the sentence (Kliegl et al., 2004). The dependent variables include numerous metrics, including fixation duration metrics such as first fixation duration (FFD) and total gaze time on a word, as well as location and numerosity metrics such as the location of a fixation on a word and gaze-regressions.

A major caveat of the computational models that have been developed since the past two decades is that they weakly address linguistic concepts beyond the level of the fixated word, with a few exceptions, such as the spillover effects related to the preview of a next word *n+1* during the current fixation on word *n* (Engbert et al., 2005). These models are also limited in their recognition of syntactic, semantic and discourse characteristics of the text due to their complexity, despite they are indispensable aspects of reading for understanding. Machine Learning (ML) models of oculomotor control address some of those limitations by presenting high predictive power. However, the holistic approach of the learning models has drawbacks in terms of the explainability of the model underpinnings. In this study, we present experiments with a traditional model of oculomotor control in reading, namely the EZ Reader (Reichle et al., 2003), two ML models (a regression model and a recurrent network model), and their combination. We present an evaluation of the results by focusing on the model inputs that reveal relatively higher accuracy.

Accordingly, the aim of the present paper is to investigate the effectiveness of both types of models and their combinations on predicting human reading behavior as set up by the CMCL 2021 shared task (Hollenstein et al., 2021). The task is defined as predicting five eye-tracking features, namely *number of fixations* (nFix), *first fixation duration* (FFD), *total reading time* (TRT), *go-past time* (GPT), and *fixation proportion* (fixProp). The eye-tracking data of the Zurich Congitive Language

Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, pages 134–140

Processing Corpus (ZuCo 1.0 and ZuCo 2.0) were used (Hollenstein et al., 2018), (Hollenstein et al., 2019). Details of these variables and further information on the data set can be found in the task description paper (Hollenstein et al., 2021).

2 Methodology

We created our models and identified the input features following the findings in research on oculomotor control in reading. The previous studies have shown that word length, frequency and sentential predictability are well known parameters that influence eye movement patterns in reading (Rayner, 1998). There exist further parameters that influence eye movement characteristics. For instance, the location of a word in the sentence has been proposed as a predictor on First Fixation Duration (Kliegl et al., 2006). Therefore, we used those additional parameters to improve the accuracy of the learning models, as well as running a traditions oculomotor control model (viz., the EZ Reader) with its required parameter set. Below we present a description of the models that have been employed in the present study.

2.1 System Description

2.1.1 The EZ Reader Model

EZ Reader has been developed as a rule-based model of oculomotor control in reading. It predicts eye movement parameters, such as single fixation duration, first fixation duration and total reading time. The model efficiently addresses some of experimental research findings in the reading literature. For example, a saccade completion takes about 20-50 msec to complete, and saccade length is about 7-9 characters (Rayner, 2009). The model consists of three main processing modules. The oculomotor system is responsible for generating and executing saccades by calculating the saccade length. The visual system controls the attention of the reader. Finally, the word identification system calculates the time for identifying a word, mainly based on the word length and the frequency of word in daily use. EZ Reader accepts four arguments as its input; frequency (count in million), word length (number of characters), sentential predictability of the word, and the word itself. The output features of the model are given in Table 1.

Among those features, FFD and TT outputs of EZ Reader directly map to FFD and TRT (Total Reading Time) in the training data of the CMCL 2021 shared task. The EZ reader output features are not sufficient enough to generate mean absolute error values for each feature in the training data. Therefore we were only able to calculate mean absolute error values for FFD and TRT. Table 2 presents the Mean Absolute Error (MAE) values of the test set, when predicted by the EZ Reader model. In the design of the EZ Reader model, we assumed the simulation count as 2000 participants, which means that the model runs 2000 distinct simulations and the result scores consist of the average of the simulation results. 2000 participants is the optimum number for our case in terms of simulation time and the MAE it produces. Above 2000 participants MEA did not decrease significantly.

A major challenge in designing the EZ Reader model is that the model is not able to produce the output values for some of the words, labeling them *Infinity*. Those are usually long words with relatively low frequency. In order to find an optimal value to fill in the *Infinity* slots, we calculated the mean absolute error between TRT of the training data and the TT values of EZ Reader model results, as an operational assumption. The calculation returned 284 ms. Figure 1 shows the MAE scores over given values between 0 to 500. This value is close to the average fixation duration for skilled readers which is about 200ms - 250ms (Rayner, 1998). Therefore, we preserved the assumption in model development pipeline.

Feature	Description
EZ-SFD	Single Fixation Duration
EZ-FFD	First Fixation Duration
EZ-GD	Gaze Duration
EZ-TT	Total Reading Time
EZ-PrF	Fixation Probability
EZ-Pr1	Probability of making exactly one fixation
EZ-Pr2	Probability of making two or more fixations
EZ-PrS	Probability of skipping

Table 1: EZ Reader output features.

EZ Reader	Training Data	MAE
TT	Total Reading Time	3.25
FFD	First Fixation Duration	9.14

Table 2: Mean Absolute Error (MAE) scores obtained by the EZ Reader model

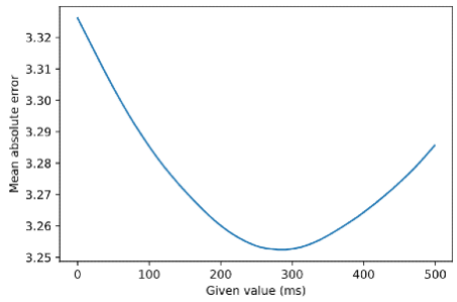

Figure 1: Mean Absolute Error scores over given values for Infinity slot

In the present study, besides calculating the mean absolute error values for the EZ Reader model, we employed the outputs of the EZ Reader model as inputs to the LSTM model. Below, we present the model for the Linear Baseline and the LSTM model.

2.1.2 Linear Baseline

Our linear model is a least-squares regression model with L2 regularization (ridge regression). The input features to the regression model include the main word-level features, frequency, word length and predictability discussed above. Word frequencies are computed using a large news corpus of approximately 2.7 million articles.[1] The predictability features are obtained using the recurrent network described in Section 2.2. Besides these features, we also include some linguistic features including the POS tag, dependency relation, and signed distance from the head, as well as named entity tag. The POS and dependency information is obtained using version 1.2 of UDPipe using the pretrained models released by the authors (Straka and Straková, 2017; Straka and Straková, 2019). We used Apache OpenNLP (Apache Software Foundation, 2014) for named entity recognition. The model input also included indicator features for beginning and end of sentence, and whether the word is combined with a punctuation mark or not (see Table 3). We also included the features from EZ-reader described in Section 2.1.1 as additional inputs to the regression model.

The predictions were based on a symmetric window around the target word, where all the above features for the target word and $\pm k$ words were concatenated. We selected the optimal window size as well as the regularization constant (alpha) for the ridge regression model via grid search. The grid search is used to determine a single same alpha and single window size for all target variables. We use the ridge regression implementation of the Python scikit-learn library (Pedregosa et al., 2011).

[1] 'All the news' data set, available from https://components.one/datasets/all-the-news-2-news-articles-dataset/.

Feature	Description	Used in model
Word Frequency (Fr)	Word occurrence per million	LB-LSTM
Word Location	Zero based index of the word in sentence.	LB-LSTM
Word Length (WL)	Character count of the word	LB-LSTM
Word Predictability (Pr)	Probability of knowing a word before reading it	LB-LSTM
StartPunct	The presence of a punctuation before the word	LB-LSTM
EndPunct	The presence of the punctuation at the end	LB-LSTM
EndSent	Is the last word of the sentence or not	LB-LSTM
POS	Core part-of-speech category	LB
Dep	Universal syntactic relations	LB
HeadDist	Signed distance from the head	LB
Ner	Named entity category (person and company names, etc.)	LB
EZ Reader simulation outputs	see Table 1	LB-LSTM

Table 3: Input features used in Linear Baseline and LSTM model.

2.1.3 LSTM Model

The LSTM model consists of an LSTM layer with 128 units followed by two dense layers and 5-dimensional output layer. The input features of the model include word length in total number of characters, word predictability, frequency per million, the location of the word in the sentence, the presence of a punctuation before the word, the presence of the punctuation at the end, and the end of sentence, being the last word of the sentence or not. Finally, the input features included the outputs of the typical EZ Reader model (given in Table 1). The output features of the LSTM model the variables identified by the CMCL 2021 share task, namely nFix, FFD, GPT, TT, and fixProp.

2.2 Predictability Scores

Sentential predictability of a word in a context is a well-established predictor of eye movement patterns in reading (Fernández et al., 2014; Kliegl et al., 2004; Clifton et al., 2016). We used two methods to generate the predictability values. First, we used the average human predictability scores from the Provo Corpus (Luke and Christianson, 2018), which is a public eye-tracking dataset collected from real participants. The Provo Corpus includes the cloze task results in which participants are given the starting word of the sentence and expected to guess the next word. The actual word is shown after the participant's guess and prediction for the next word is expected. This process continues for all of the words. Prediction value is generated for each word in corpus by simply calculating the ratio of the correct guesses to all guesses for the word. We selected 1.0 as the default prediction value for words which does not exist in the

Model	nFix	FFD	GPT	TRT	fixProp
LAST	3.88	0.66	2.20	1.52	10.81
Linear	4.36	0.74	2.50	1.76	12.55
LSTM	4.62	0.76	3.61	1.84	13.06
Baseline	7.30	1.15	3.78	2.78	21.78

Table 4: Official scores (MAE) of our models in comparison to mean baseline and the first team (LAST) in the competition.

Features	nFix	FFD	GPT	TRT	fixProp
Fr	4.80	2.20	2.75	1.85	13.61
WL	6.73	0.77	2.78	1.84	12.94
Pr	5.64	0.85	3.11	2.15	15.17
EZ-SFD	6.26	1.00	3.11	2.34	18.21
WL x Pr x Fr	4.35	0.71	2.68	1.73	11.99
WL x Pr	4.28	0.71	2.70	1.68	12.07
EZ-SFDxFrxWLxPr	4.21	0.73	2.57	1.64	12.11

Table 5: MAE for with different feature combinations.

Provo Corpus. The mean absolute error for TRT between EZ Reader output and CMCL train data was at minimum when default prediction value is 1.0.

Second, we developed a separate LSTM model that produced sentential predictability values. For this, we trained the model by Wikipedia.[2] Since the primary goal of the model was to predict eye movement patterns per word, we built a word-level language model. The model consisted of two LSTM layers with 1200 hidden units. It was trained with a learning rate of 0.0001, and a dropout value was set to 0.2, with the Adam optimizer. After the training, we obtained the predictability scores for each word based on their sentential context. These scores were then used as an additional feature in our final model besides the other features, such as word length and frequency.

Provo Corpus predictability values are independent from the context of text used in the shared task. However using predictability values from the first method gave better results than the calculated predictability. Therefore we used Provo Corpus predictability values for the results in the following sections.

3 Results

We participated in the CMCL 2021 shared task with two submissions, one with the linear model described in Section 2.1.2, and the other with the LSTM model (Section 2.1.3). Table 4 presents the scores of our system in the competition, in comparison to mean baseline and the best system. Our systems perform substantially better than the baseline, and the difference between the scores of the participating teams are comparatively small. Among our models, the linear model performed slightly better, obtaining 10th place in the competition. However, experimenting with the LSTM model gave us more information about the basic features of eye movements in reading and their effects on fixation durations. For the remainder of this section, we will present further experiments with the LSTM model, demonstrating the effects of various features discussed above.

3.1 Further Experiments

To demonstrate the effectiveness of the features described above, we trained a number of models that employed a set of input variables in isolation, as well as the models trained by their combination. In particular, we trained a model on frequency, then predictability, and then word length. Then we trained models by their combinations as input features. Each model produced a MAE (mean absolute error) value. We then calculated the average of the MAE values for each model output (nFix, FFD, GPT, TRT, and FixProp) and their Standard Deviation (SD). Finally, we calculated how far each model was away from the average MAE in terms of the SDs. Table 5 presents MAE scores for each setting.

The figures in the Appendix A show the distance of the models from the center of the circle, where the center represents the best MAE score and the circle represents the distance covered by one SD (Standard Deviation) from the best accuracy (i.e. the center). The models that received the combination of frequency, predictability, word length and E-Z SFD (i.e., E-Z Reader's single fixation duration prediction) as the input returned the best MAE values for four of five dependent variables. As an example, consider the MAE values for the models developed for predicting the *nFix* (the number of fixations on a word). Figure 2 shows that the majority of the models that are based on features in isolation have relatively lower predictability compared to the models that take a combination of the features as the inputs. In particular, the predictability model (i.e., the model that is solely based on

[2]We use the sentence segmented corpus from https://www.kaggle.com/mikeortman/wikipedia-sentences.

the predictability values as the input feature) has a mean MAE value 1.75 times the SD (Standard Deviation). Similarly, the word-length model has approximately 3 times the SD from the best score, and the EZ-SFD model (i.e., the model that is solely based on the single fixation duration predictions of the EZ Reader model) has a mean MAE value far away from the mean by 2.5 times the SD value.

4 Conclusion

In this paper, we analyzed a linear baseline model and an LSTM model that employed the outputs of a traditional model as its inputs. We built models with input features in isolation, and their combination. The evaluation of the mean absolute errors (MAE) supported a major finding in reading research: The oculomotor processes in reading are influenced by multiple factors. Temporal and spatial aspects of eye movement control are determined by linguistic factors as well as low-level nonlinguistic factors (Rayner, 1998; Kliegl and Engbert, 2013). The models that employ their combinations return higher accuracy. Our findings also indicate that besides the frequently used features in the literature, the EZ-SFD (single frequency duration outputs of the EZ Reader model) may contribute to the accuracy of the learning based models. Nevertheless, given the high variability of the machine learning model outputs a systematic investigation is necessary that address several operational assumptions in the present study. In particular, future research should improve statistical analysis for comparing the model outputs. It should also address the influence of the location of a word in a sentence, besides its interaction with the duration measures. Last but not the least, future research on developing ML models of oculomotor control in reading should focus on the relationship between the aspects of the ML model design and basic findings in reading research. The GCMW (Gaze Contingent Moving Window) paradigm and the boundary paradigm (Rayner, 2014) are some examples of those findings that could be used for oculomotor control modeling.

References

Apache Software Foundation. 2014. openNLP Natural Language Processing Library. http://opennlp.apache.org/.

Charles Clifton, Fernanda Ferreira, John M. Henderson, Albrecht W. Inhoff, Simon P. Liversedge, Erik D. Reichle, and Elizabeth R. Schotter. 2016. Eye movements in reading and information processing: Keith Rayner's 40 year legacy. *Journal of Memory and Language*, 86:1–19.

Ralf Engbert, Antje Nuthmann, Eike M Richter, and Reinhold Kliegl. 2005. SWIFT: a dynamical model of saccade generation during reading. *Psychological review*, 112(4):777.

Gerardo Fernández, Diego E. Shalom, Reinhold Kliegl, and Mariano Sigman. 2014. Eye movements during reading proverbs and regular sentences: The incoming word predictability effect. *Language, Cognition and Neuroscience*, 29(3):260–273.

Nora Hollenstein, Emmanuele Chersoni, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. Cmcl 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading. *Scientific data*, 5(1):1–13.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2019. Zuco 2.0: A dataset of physiological recordings during natural reading and annotation. *arXiv preprint arXiv:1912.00903*.

Reinhold Kliegl and Ralf Engbert. 2013. Evaluating a computational model of eye-movement control in reading. In *Models, simulations, and the reduction of complexity*, pages 153–178. De Gruyter.

Reinhold Kliegl, Ellen Grabner, Martin Rolfs, and Ralf Engbert. 2004. Length, frequency, and predictability effects of words on eye movements in reading. *European journal of cognitive psychology*, 16(1-2):262–284.

Reinhold Kliegl, Antje Nuthmann, and Ralf Engbert. 2006. Tracking the mind during reading: The influence of past, present, and future words on fixation durations. *Journal of experimental psychology: General*, 135(1):12.

Steven G Luke and Kiel Christianson. 2018. The provo corpus: A large eye-tracking corpus with predictability norms. *Behavior research methods*, 50(2):826–833.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

Keith Rayner. 2009. Eye Movements in Reading: Models and Data. *Journal of Eye Movement Research*, 2(5):1–10.

Keith Rayner. 2014. The gaze-contingent moving window in reading: Development and review. *Visual Cognition*, 22(3-4):242–258.

Erik D Reichle, Keith Rayner, and Alexander Pollatsek. 2003. The EZ reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and brain sciences*, 26(4):445.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2019. Universal dependencies 2.5 models for UDPipe (2019-12-06). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Appendix

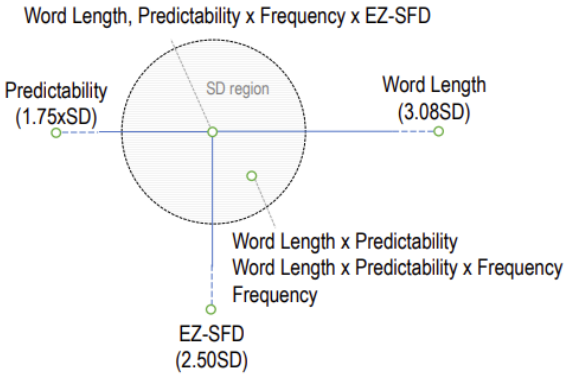

Figure 2: MAE scores for nFix

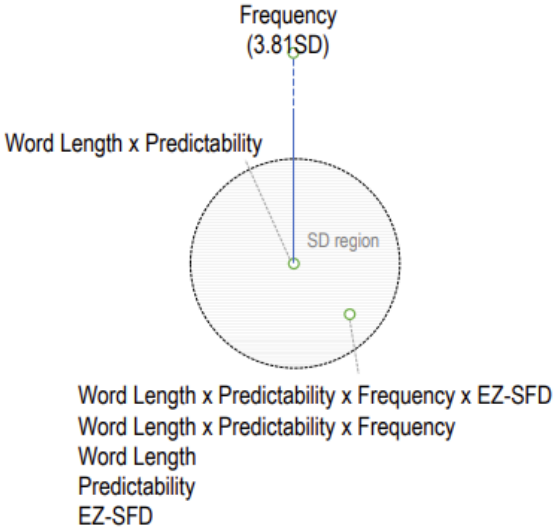

Figure 3: MAE scores for FFD

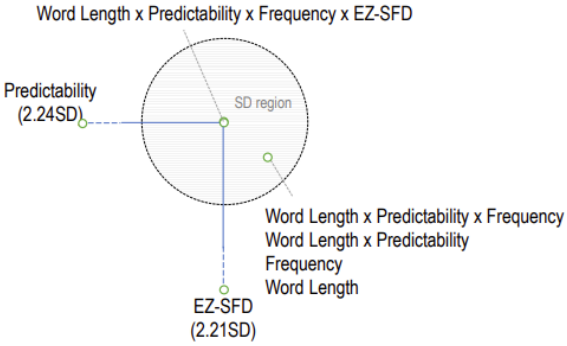

Figure 4: MAE scores for GPT

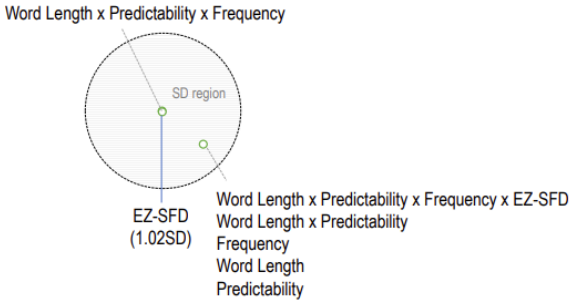

Figure 5: MAE scores for fixProp

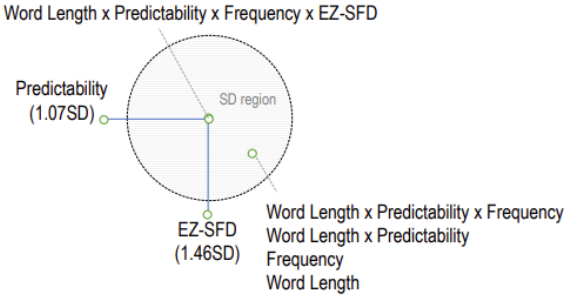

Figure 6: MAE scores for TRT