Project Report

on

PREDICTING READING COMPREHENSION FROM GAZE BEHAVIOUR

Report submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology

in

Mechanical Engineering

Submitted by

Gollu Jikki Sravanthi

(19ME31010)

Under the guidance of

Prof. Manjira Sinha

CENTRE FOR EDUCATIONAL TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

YEAR 2022-2023

DECLARATION

We certify that

a) The work contained in this report has been done by us under the guidance of our supervisor.

b) The work has not been submitted to any other Institute for any degree or diploma.

c) We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d) Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Date: November, 2022 (Gollu Jikki Sravanthi)

Place: Kharagpur (19ME31010)

CERTIFICATE

This is to certify that the project entitled, “Predicting Reading Comprehension from Gaze Behavior”, submitted by Gollu Jikki Sravanthi (19ME31010) is a record of bonafide research work carried out by her in the Centre for Educational Technology, Indian Institute of Technology Kharagpur under my supervision and guidance for the partial fulfillment for the award of degree of Bachelor of Technology in Mechanical Engineering during academic session 2022-2023 from Indian Institute of Technology, IIT Kharagpur.

Prof. Manjira Sinha

(Assistant Professor)

Centre for Educational technology

Indian Institute of Technology

E-mail: manjira.sinha@cet.iitkgp.ac.in

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Manjira Sinha for her tremendous help, support and guidance throughout the work. I deeply express my sincere thanks for encouraging and allowing me to work on the project under her able guidance. She has not only suggested the problem but also helped me in various other aspects of the project and provided me with the opportunities to expand my knowledge and experience. This work has been possible because of her inspiration, motivation and full freedom given to me to incorporate any ideas.

I am sincerely grateful to the Head of the Centre for Educational technology, Indian Institute of Technology, Kharagpur for providing all the necessary facilities for the successful carryout of the project.

Gollu Jikki Sravanthi

(19ME31010)

Fourth year Undergraduate student

Department of Mechanical Engineering

Indian Institute of Technology, Kharagpur

TABLE OF CONTENTS

INTRODUCTION

People generally move their eyes in a prevailing way while reading a text or responding to a question. There may be a self-similar behavior in every individual, where they act differently based on their understanding of the text or passage. This model attempts to decode the individual's behavior from the readings of their eye movements like fixation duration and more. It is known from observations that eye movements can reflect cognitive processes like viewing, searching, reading, and more. There are many types of research regarding mental state from eye-tracking data. Most of them included where people have to predict their unassigned tasks or an individual's characteristics, i.e., their personality. Only a few are focused on predicting their state during reading and their level of understanding of the text read. Here, deep networks like CNN and RNN are used to expect the reader's difficulty level and nativity from eye fixations and fixation durations.

Different approaches are tried to classify the related variables, the difficulty of reading, and whether the reader's first language is English. Fixation data is collected from eye-tracking data of 95 participants who read SAT practice passages. To find the level of comprehension, the participants are asked to respond to different comprehension questions followed by self-evaluation questions. Here it is discovered only the difficulty level from the fixation data without references to levels of comprehension.

The performances of the leading models used: Convolutional Neural Networks (CNN) and Recurrent Neural Networks, along with side models, and Null Accuracy (without any model) are compared. Our main goal was to predict the reader's comprehension levels during reading from the fixation data obtained from their eye-tracking data. If this method proves successful, it can help evaluate the reader's comprehension without manually answering the testing questions. This model can help create intelligent systems that can give the mental state of a reader instantaneously. Thus, the model allows us to estimate the relationship between reading behavior and comprehension of a person from their reading patterns and explains excellent or bad understanding capability.

LITERATURE REVIEW

Many studies and research were done to estimate from the eye-tracking data. Boisvert and Bruce 2016 and Henderson et al. 2013 focused on predicting the tasks like viewing or memorization, etc. Al-Samarraie et al. 2017, 2018 are based on an individual's personality estimation.

It is found by Underwood et al. 1990 studies (one of the first studies to predict reading comprehension from eye tracking data) that fixation duration helps in predicting the difficulty or comprehension level of a person reading the text. The model was trained and tested on their dataset but could not give generalizations over unknown or new datasets.

Some other studies focused on finding the literacy of an individual, Augereau et al. 2016 and Lou et al. 2017. They were able to find the literacy skill of an individual, but that is not related to our current model because that model is not based on eye movement or fixation measurements. In another study by Makowski et al. 2018, estimations of comprehension levels from eye-tracking data are done, but it still needs to produce better results.

Thus, whether a model could predict comprehension levels remains a question. It is still to be known if it can produce optimistic results. Therefore, here it is tried to classify our data into two variables, the difficulty of reading and nativity, and try to estimate the accuracies by the neural network's models.

DATA

The SAT dataset is one of the most critical and largest eye-tracking data of fixations. The eye movements of 95 participants reading four SAT passages are recorded. It also consists of answers to comprehension and self-evaluation questions (like the difficulty of understanding) of every individual after reading the passages. The passages, questions, and data are collected from the SAT dataset available publicly at https://github.com/ahnchive/SB-SAT. Participants were asked to read the text from the passages displayed one at a time to analyze the reading and answer the comprehension questions. A 19-inch flat-screen CRT ViewSonic SVGA monitor was used to display the passages for reading. Five minutes are given to read the passage, and no time limit is held for answering the questionnaire. The eye positions were recorded using an Eye Link 1000 sampling. The fixations were parsed from the gaze coordinates.

Here, the model tries to find the accuracy of the difficulty of understanding and whether they are native. Thus, the comprehension questions, the answers, and the levels of overall comprehension are not calculated. It is focused on the sequences of 21 fixation-location (x, y), fixation duration, and features of the pupil size extracted from the dataset. These recorded values are sent as inputs for the current model to predict an individual's difficulty level of reading and nativity.

The responses of the participants decide the level of difficulty of the passages. They were asked to rate the difficulty where the passages are considered to be of a high level if the participants rated them as hard or very hard. They are considered to be of a low level if the participants rated them as easy or very easy. The final data is arranged based on whether the participant is a native speaker of English.

It is observed that people with higher understanding levels made shorter fixation durations and had a faster reading rate. The people who rated the passage hard or very hard tended to read slowly and had larger fixation durations than others. It is also observed that native speakers also took smaller fixation durations and had a high reading rate. There were few changes in the observations with the pupil sizes. It showed no difference across the levels. Thus, these results jointly show the levels of understanding through different sequences of eye movements and their behavior during reading.

PRE-PROCESSING

There are many samples of data in the dataset for every individual. The labels of the datasets are stored, and the corresponding information from the tables is changed to 0 or 1, according to the half they belong, to separate the high and low levels. The main four oculomotor features - Current fixation of X, Current fixation of y, Current fixation of Pupil, and Current Fixation Duration from each data are normalized to values between 0 and 1. The data retrieved is grouped into a window of 21 consecutive fixations and fixation durations, without overlaps, to pass them as inputs for the model training. The dataset has to be labeled for differentiation and to address the results of every individual separately. But here, labeling could become a considerable problem if done one by one during compilation or manually, as the dataset is vast, to begin with. So, the labels for the data are inherited directly from the label of the passage from which the window came and label of that particular individual and are labeled as high and low. It is assumed that reading at a local scale is the same as reading at a more global scale.

```
msd001    4
msd002    4
msd003    4
msd004    4
msd005    4
         ..
msd102    4
msd103    4
msd105    4
msd106    4
msd107    4
Name: subj, Length: 95, dtype: int64

dickens      95
flytrap      95
genome       95
northpole    95
```

```
Name: book, dtype: int64

1    231
0    149
Name: acc_level, dtype: int64

1    180
0    200
Name: subj_acc_level, dtype: int64

1    161
0    219
Name: confidence, dtype: int64

0    252
1    128
Name: difficulty, dtype: int64

0    291
1     89
Name: familiarity, dtype: int64

0    373
1      7
Name: recognition, dtype: int64

1    230
0    150
Name: interest, dtype: int64

0    250
1    130
Name: pressured, dtype: int64
```

```
0    233
1    147
Name: sleepiness, dtype: int64

1    200
0    180
Name: sleephours, dtype: int64

1    252
0    128
Name: sex, dtype: int64

1    264
0    116
Name: native, dtype: int64
```

The data is taken in order of the individual's label and then in order of the passage they read. After slicing the four primary columns from the original eye-tracking data, they are assigned the individual's name and the passage using

the function generate_window. All these values are stored as a tuple of 21 consecutive points, which represent the 21 fixations. The predictions from the fixations are made in three different manners: prediction of new reading fixation windows, fixation windows on new passages, and fixation windows on new subjects.

Prediction on New Reading Windows:

New windows mean the model has just seen those data. So, the model did not work on this data before, and it has to predict the output for this new set of windows. Thus, the original dataset of 11,548 samples is randomly divided into three proportions. The proportions are 60% for training, 20% for validation, and 20% for testing datasets.

Prediction on New Passages:

This method is to predict if the model could generalize from new passages. This method tries to predict the comprehension level of a reader for an unseen passage based on the training from the other two datasets. So, the dataset is divided again into three proportions, where two passages are randomly allocated for the training dataset, one passage for validation, and one passage is left for testing the model. Random splits are generated for different readers, so there would be less chance of two or more readers getting the same passages in the dataset.

Prediction on New Readers:

This method tests whether the model could be generalized for new readers. The preliminary dataset of 95 participants is again split into three proportions. 60% of 95 participants, i.e., 57 participants' windows, are included in the training dataset, 20%, i.e., 19 participants' windows in the validation dataset, and the rest, 19 participants' windows in the testing dataset. The comprehension levels for the unseen 19 participants are predicted after training.

The values in every dataset are joined together into an array, and thus, three new datasets are formed using the function create_dataset. The information of every data is stored in three data frames to access whenever needed. The models for the datasets are saved, and the data frames are copied into a CSV file for future purposes.

MODEL

After inputting the datasets for training, validation, and testing, two levels of a predicted variable, a high or low level of comprehension are found.

Null Model:

The Null model is calculated for a lower bound. It simply returns the outputs based on the weights of the classes, i.e., how frequently the required class occurs in a dataset. It calculates the weights of each class (0 or 1) in the training dataset and outputs them along with the loss weight. The null accuracy is calculated from the maximum occurrence of the weights in the testing dataset.

- The networks are built using the sequential model. It is a model representing a plain stack of layers; each layer has exactly one input tensor and one output tensor.

Convolutional Neural Networks (CNN):

The CNN network has three one-dimensional convolutional layers and two full-connected layers. The convolutional layers are the primary building block of a CNN. The layer retains a collection of filter parameters learned step by step during the training. All three convolutional layers have 40 filters with size three and stride 1, without zero-padding, followed by an activation layer. After these six layers, it is followed by a MaxPooling1D layer to downsample the input representation by pool_size of 2. A Dropout layer is added, where randomly selected neurons are ignored during training so that the network becomes less sensitive to specific weights. Then, it continued with a Flatten layer to flatten the input without affecting the batch size. A dense layer, then activation, dropout, and again dense, followed by activation layers, are created. The consequent feature maps are fully connected to 50 units, again connected to 20 units. At last, a SoftMax layer is added, which outputs the units corresponding to the two levels of the predicted variable by converting the vector of values into a probability distribution.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 19, 40)            520

 activation (Activation)     (None, 19, 40)            0

 conv1d_1 (Conv1D)           (None, 17, 40)            4840

 activation_1 (Activation)   (None, 17, 40)            0

 conv1d_2 (Conv1D)           (None, 15, 40)            4840

 activation_2 (Activation)   (None, 15, 40)            0

 max_pooling1d (MaxPooling1D  (None, 7, 40)            0
 )

 dropout (Dropout)           (None, 7, 40)             0

 flatten (Flatten)           (None, 280)               0

 dense (Dense)               (None, 50)                14050

 activation_3 (Activation)   (None, 50)                0

 dropout_1 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 20)                1020

 activation_4 (Activation)   (None, 20)                0

 dense_2 (Dense)             (None, 2)                 42

=================================================================
Total params: 25,312
Trainable params: 25,312
Non-trainable params: 0
```

Recurrent Neural Network (RNN):

A Bidirectional LSTM layer is used here for better performance as LSTM (Long Short-Term Memory) helps to remember the inputs for a long time and could use when necessary. Bidirectional trains two LSTMs instead of one for quicker results and complete learning on the problem. The model consists of two bidirectional layers, where each LSTM cell consists of 25 hidden units. The return sequence is marked true so that the output of the hidden state is used as input to the next LSTM layer. A dropout layer follows it. The rest is done the same way as it is for the CNN model. A dense layer, activation, dropout, dense and activation layers are included. At last, a dense layer with SoftMax is used to output the units of the corresponding predicted variables.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 21, 50)           6000
 l)

 bidirectional_1 (Bidirectio  (None, 50)               15200
 nal)

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 50)                2550

 activation (Activation)     (None, 50)                0

 dropout_1 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 20)                1020

 activation_1 (Activation)   (None, 20)                0

 dense_2 (Dense)             (None, 2)                 42

=================================================================
Total params: 24,812
Trainable params: 24,812
Non-trainable params: 0
_________________________________________________________________
None
```

The final step of building the model is the compilation. The loss function finds the error or deviation in the learning process. It takes two arguments, y-true (the actual labels) and y-pred (prediction with a dataset of the same shape as y_true,i.e., valid dataset). Categorical_crossentrophy is considered because the model needs to output two output variables of 0s and 1s. Adam optimizer optimizes the input weights by comparing loss and prediction functions. The metric evaluates the performance of the model based on accuracy. The accuracy is found from the y_true and y_pred arguments from training and valid datasets, respectively. ModelCheckPoint is a callback function used while running model.fit(). It saves a model or weights at some interval so the model can be loaded again later from the saved point. Here, the model is saved only when the model is best according to the metric accuracy of valid dataset. The training is stopped when the metric loss function of the valid dataset stops changing even after running 50 epochs. Model.fit() is called, which returns a history object that contains loss and accuracy values of training and validation datasets, respectively.

CONCLUSION:

The plots consist of

training accuracy vs validation accuracy, training loss vs validation loss

confusion matrix for testing dataset and result of the model

PREDICTING ON NEW WINDOWS – RECORD WISE

NULL ACCURACY

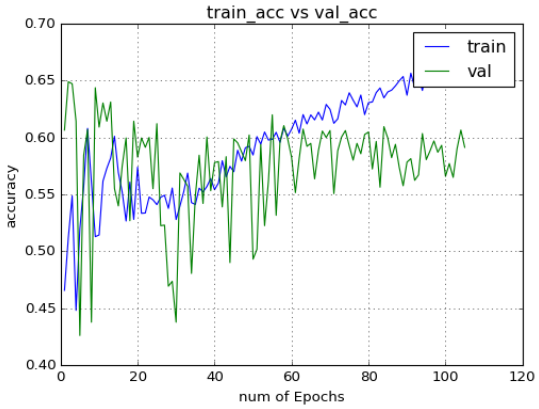
DIFFICULTY acc=0.67%

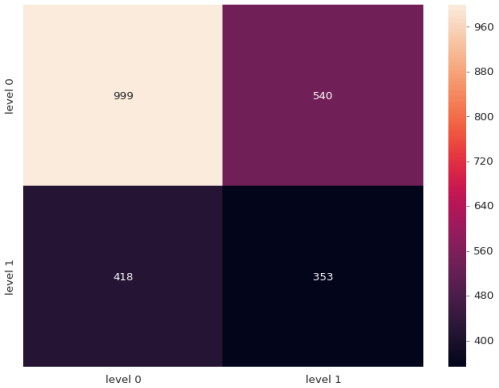
```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    6928
# of samples for validation is:  2310
# of samples for prediction is:  2310
# of total sampels:      11548

##### data imbalances #####
0    0.661374
1    0.338626
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.7560017459624618, 1: 1.4765558397271952}

##### null acc for test dataset #####
0.6662337662337663
```

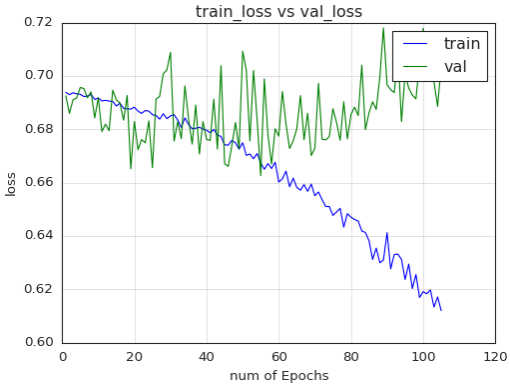
NATIVE acc=0.66%

```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    6928
# of samples for validation is:  2310
# of samples for prediction is:  2310
# of total sampels:      11548

##### data imbalances #####
0    0.346709
1    0.653291
Name: native, dtype: float64

##### loss weight #####
{0: 1.4421315570358035, 1: 0.7653557224922669}

##### null acc for test dataset #####
0.6571428571428571
```

CNN for DIFFICULTY acc=0.59%

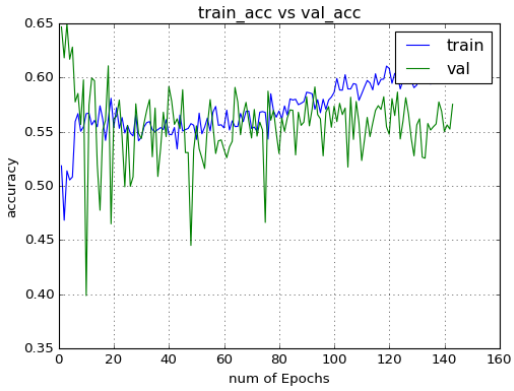


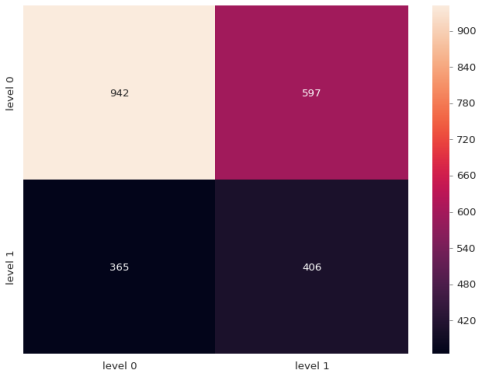





```
predicted variable: difficulty

Test Loss: 0.7041446566581726
Test accuracy: 0.5852813720703125
73/73 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     level 0       0.71      0.65      0.68      1539
     level 1       0.40      0.46      0.42       771

    accuracy                           0.59      2310
   macro avg       0.55      0.55      0.55      2310
weighted avg       0.60      0.59      0.59      2310

confusion matrix:
 [[999 540]
 [418 353]]
Balanced acc score: 0.55348487951396
Balanced error rate: 0.4465151204860403
```

RNN for DIFIICULTY acc=0.58%

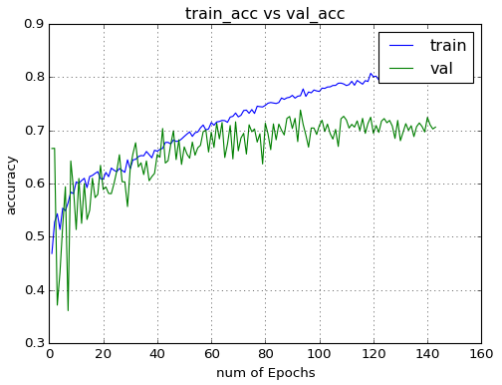


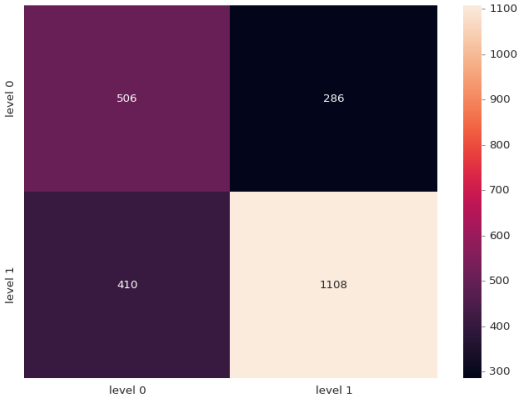



```
predicted variable: difficulty

Test Loss: 0.6729407906532288
Test accuracy: 0.5835497975349426
73/73 [==============================] - 3s 9ms/step
              precision    recall  f1-score   support

     level 0       0.72      0.61      0.66      1539
     level 1       0.40      0.53      0.46       771

    accuracy                           0.58      2310
   macro avg       0.56      0.57      0.56      2310
weighted avg       0.62      0.58      0.59      2310

confusion matrix:
 [[942 597]
 [365 406]]
Balanced acc score: 0.5693373078177502
Balanced error rate: 0.43066269218224984
```

Both RNN and CNN accuracies are near Null Accuracy but below it.

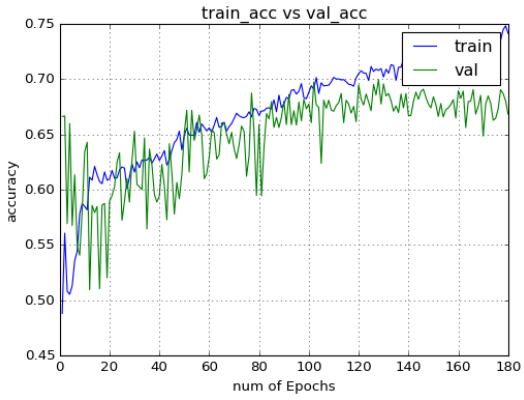
CNN for NATIVE acc=0.70%

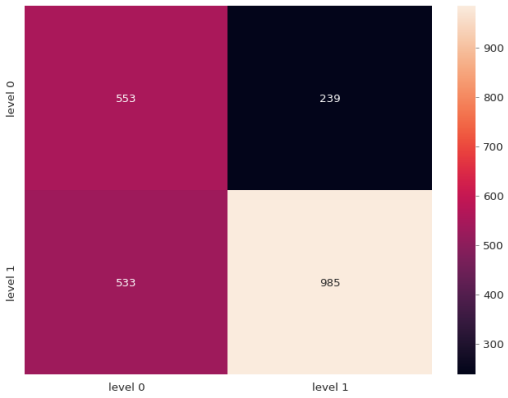


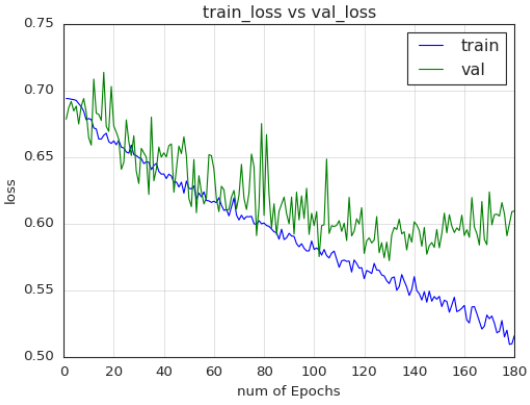



```
predicted variable: native

Test Loss: 0.6426030993461609
Test accuracy: 0.6987013220787048
73/73 [==============================] - 0s 2ms/step
              precision    recall  f1-score   support

     level 0       0.55      0.64      0.59       792
     level 1       0.79      0.73      0.76      1518

    accuracy                           0.70      2310
   macro avg       0.67      0.68      0.68      2310
weighted avg       0.71      0.70      0.70      2310

confusion matrix:
 [[ 506  286]
 [ 410 1108]]
Balanced acc score: 0.6843983311374615
Balanced error rate: 0.3156016688625385
```

RNN for NATIVE acc=0.67%








```
predicted variable: native

Test Loss: 0.614960789680481
Test accuracy: 0.6658008694648743
73/73 [==============================] - 3s 9ms/step
              precision    recall  f1-score   support

     level 0       0.51      0.70      0.59       792
     level 1       0.80      0.65      0.72      1518

    accuracy                           0.67      2310
   macro avg       0.66      0.67      0.65      2310
weighted avg       0.70      0.67      0.67      2310

confusion matrix:
 [[553 239]
 [533 985]]
Balanced acc score: 0.6735562143170839
Balanced error rate: 0.3264437856829161
```

Both CNN and RNN accuracies are more than or equal to null accuracy.

PREDICTING ON NEW PASSAGES – BOOK WISE

NULL ACCURACY

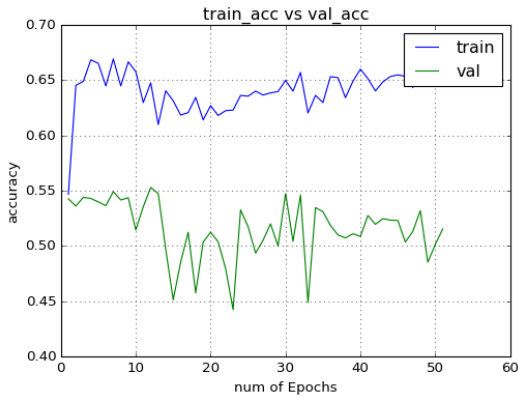
DIFFICULTY acc=0.69%

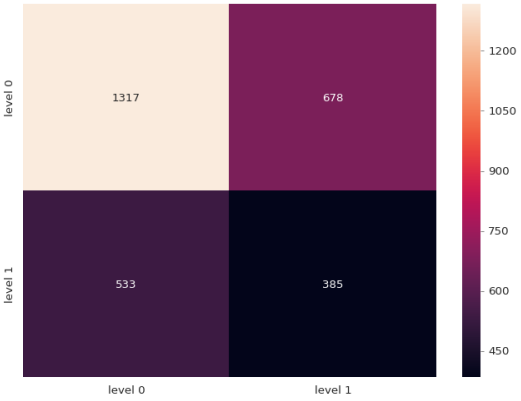
```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    5723
# of samples for validation is:  2912
# of samples for prediction is:  2913
# of total sampels:      11548

##### data imbalances #####
0    0.684431
1    0.315569
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.7305335716109267, 1: 1.584440753045404}

##### null acc for test dataset #####
0.6848609680741503
```

NATIVE acc=0.64%

```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    5723
# of samples for validation is:  2912
# of samples for prediction is:  2913
# of total sampels:      11548

##### data imbalances #####
0    0.331819
1    0.668181
Name: native, dtype: float64

##### loss weight #####
{0: 1.5068457082675093, 1: 0.7483002092050209}

##### null acc for test dataset #####
0.6392035702025404
```

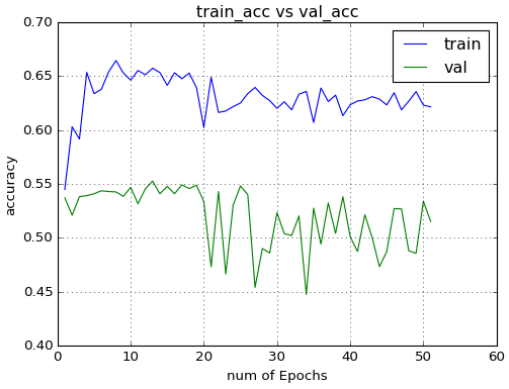
CNN for DIFFICULTY acc=0.58%

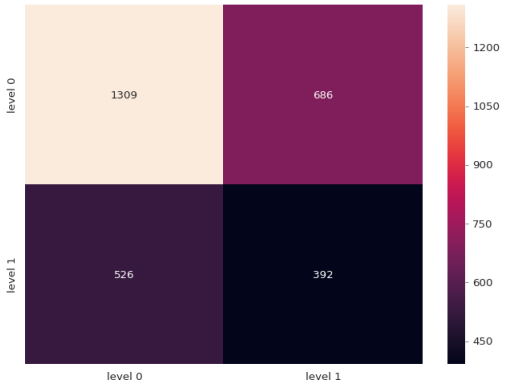


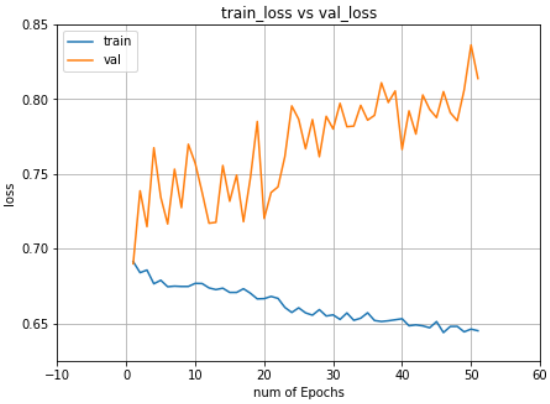



```
predicted variable: difficulty

Test Loss: 0.7251739501953125
Test accuracy: 0.5842773914337158
92/92 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     level 0       0.71      0.66      0.69      1995
     level 1       0.36      0.42      0.39       918

    accuracy                           0.58      2913
   macro avg       0.54      0.54      0.54      2913
weighted avg       0.60      0.58      0.59      2913

confusion matrix:
 [[1317  678]
 [ 533  385]]
Balanced acc score: 0.5397701770766786
Balanced error rate: 0.4602298229233214
```

RNN for DIFFICULTY acc=0.58%



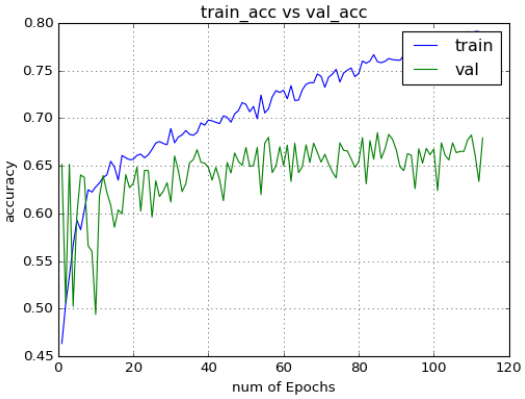


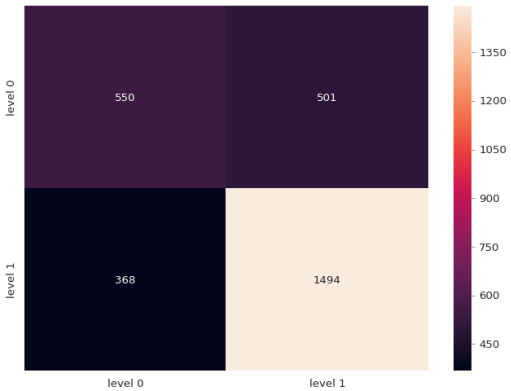



```
predicted variable: difficulty

Test Loss: 0.7216569781303406
Test accuracy: 0.5839340686798096
92/92 [==============================] - 4s 13ms/step
              precision    recall  f1-score   support

     level 0       0.71      0.66      0.68      1995
     level 1       0.36      0.43      0.39       918

    accuracy                           0.58      2913
   macro avg       0.54      0.54      0.54      2913
weighted avg       0.60      0.58      0.59      2913

confusion matrix:
 [[1309  686]
 [ 526  392]]
Balanced acc score: 0.5415778007109276
Balanced error rate: 0.45842219928907235
```

Both CNN and RNN perform poorly below Null Accuracy

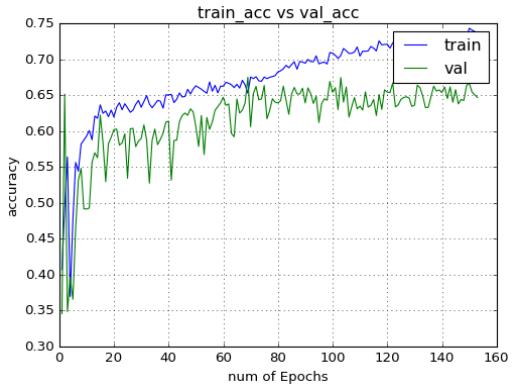
CNN for NATIVE acc=0.70%

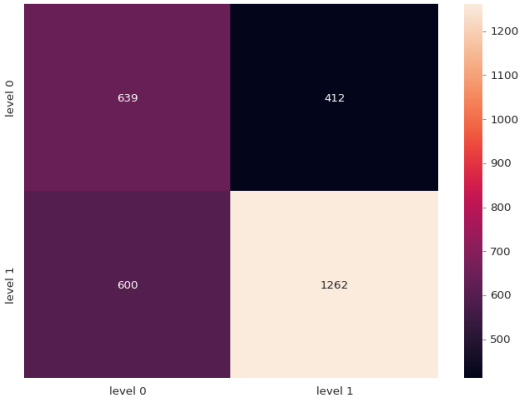





```
predicted variable: native

Test Loss: 0.6139844655990601
Test accuracy: 0.7016820907592773
92/92 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     level 0       0.60      0.52      0.56      1051
     level 1       0.75      0.80      0.77      1862

    accuracy                           0.70      2913
   macro avg       0.67      0.66      0.67      2913
weighted avg       0.69      0.70      0.70      2913

confusion matrix:
 [[ 550  501]
 [ 368 1494]]
Balanced acc score: 0.6628370913691732
Balanced error rate: 0.33716290863082676
```

RNN for NATIVE acc=0.65%






```
predicted variable: native

Test Loss: 0.6413869857788086
Test accuracy: 0.6525918245315552
92/92 [==============================] - 3s 8ms/step
              precision    recall  f1-score   support

     level 0       0.52      0.61      0.56      1051
     level 1       0.75      0.68      0.71      1862

    accuracy                           0.65      2913
   macro avg       0.63      0.64      0.64      2913
weighted avg       0.67      0.65      0.66      2913

confusion matrix:
 [[ 639  412]
 [ 600 1262]]
Balanced acc score: 0.6428791156905449
Balanced error rate: 0.3571208843094551
```

Both CNN and RNN perform well and best predicted by CNN

PREDICTING ON NEW READERS – SUBJECT WISE

NULL ACCURACY

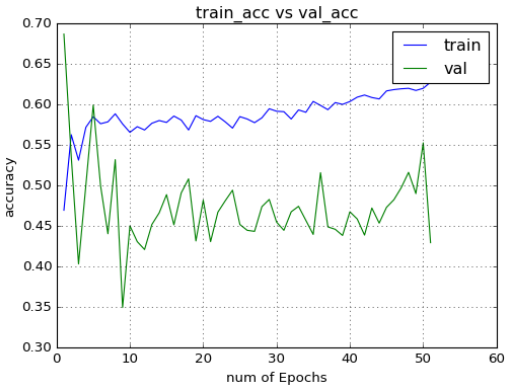
DIFFICULTY acc=0.77%

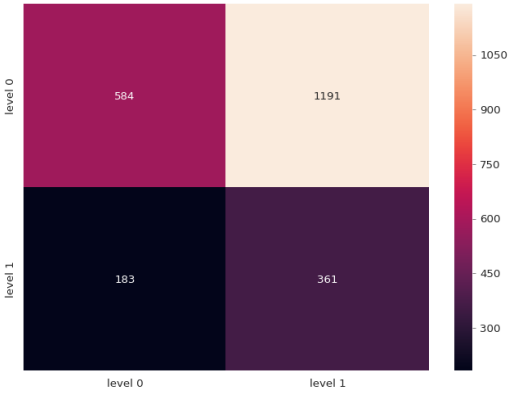
```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    6867
# of samples for validation is:  2362
# of samples for prediction is:  2319
# of total sampels:      11548

##### data imbalances #####
0    0.615116
1    0.384884
Name: difficulty, dtype: float64

##### loss weight #####
{0: 0.8128551136363636, 1: 1.2990919409761634}

##### null acc for test dataset #####
0.7654161276412247
```

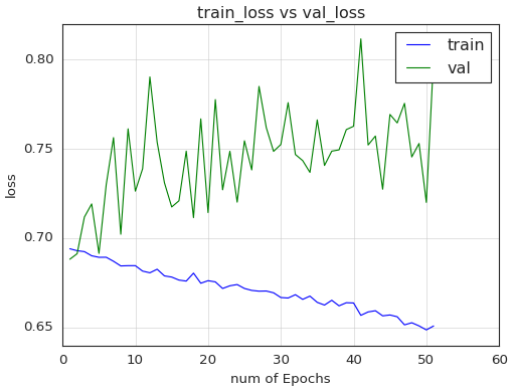
NATIVE acc=0.77%

```
##### data description #####
# of classes:    2
input shape is:  (21, 4)
# of samples for training is:    6867
# of samples for validation is:  2362
# of samples for prediction is:  2319
# of total sampels:      11548

##### data imbalances #####
0    0.385612
1    0.614388
Name: native, dtype: float64

##### loss weight #####
{0: 1.2966389728096677, 1: 0.8138184403887178}

##### null acc for test dataset #####
0.7705907718844329
```

CNN for DIFFICULTY acc=0.41%

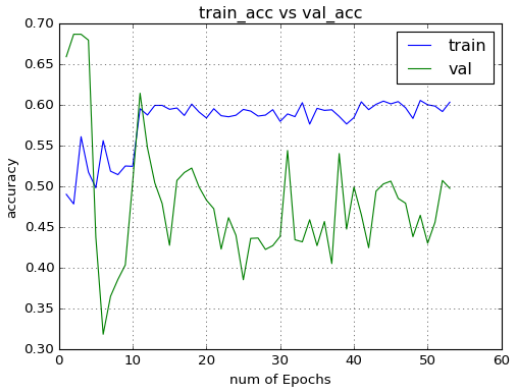


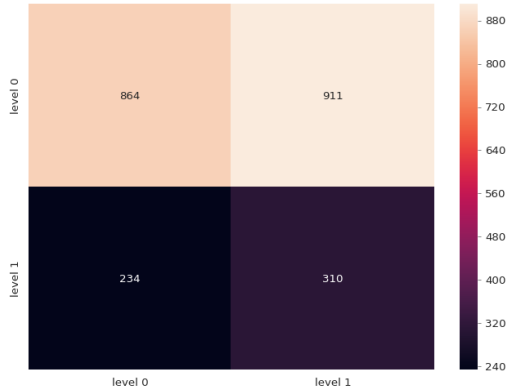


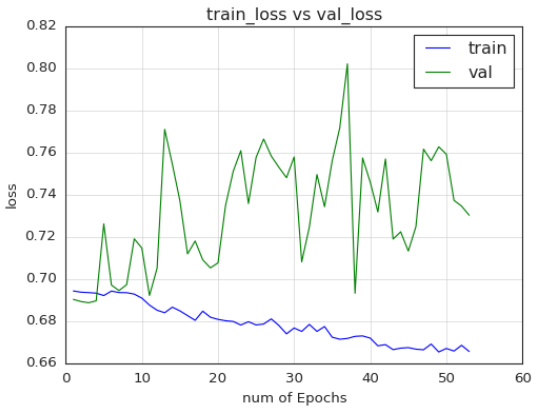



```
predicted variable: difficulty

Test Loss: 0.828581690788269
Test accuracy: 0.40750324726104736
73/73 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     level 0       0.76      0.33      0.46      1775
     level 1       0.23      0.66      0.34       544

    accuracy                           0.41      2319
   macro avg       0.50      0.50      0.40      2319
weighted avg       0.64      0.41      0.43      2319

confusion matrix:
 [[ 584 1191]
 [ 183  361]]
Balanced acc score: 0.49630851284175637
Balanced error rate: 0.5036914871582436
```

RNN for DIFFICULTY acc=0.51%



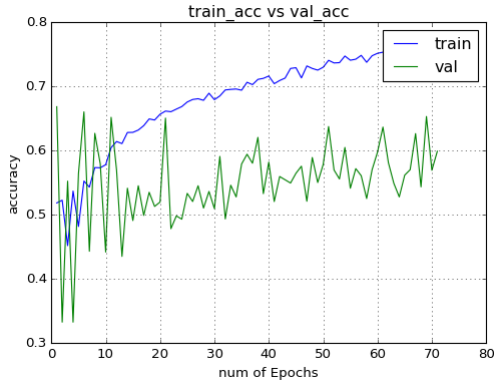


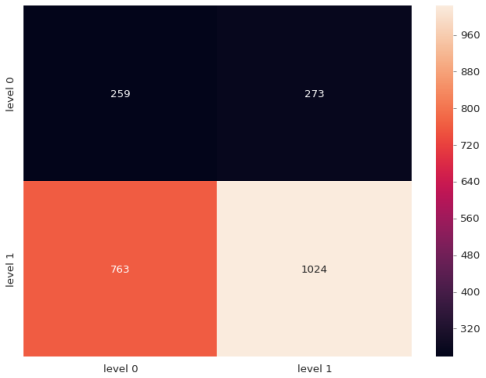



```
predicted variable: difficulty

Test Loss: 0.7214058637619019
Test accuracy: 0.5062527060508728
73/73 [==============================] - 3s 9ms/step
              precision    recall  f1-score   support

     level 0       0.79      0.49      0.60      1775
     level 1       0.25      0.57      0.35       544

    accuracy                           0.51      2319
   macro avg       0.52      0.53      0.48      2319
weighted avg       0.66      0.51      0.54      2319

confusion matrix:
 [[864 911]
 [234 310]]
Balanced acc score: 0.5283067522783761
Balanced error rate: 0.4716932477216239
```

Both CNN and RNN perform poorly below Null Accuracy

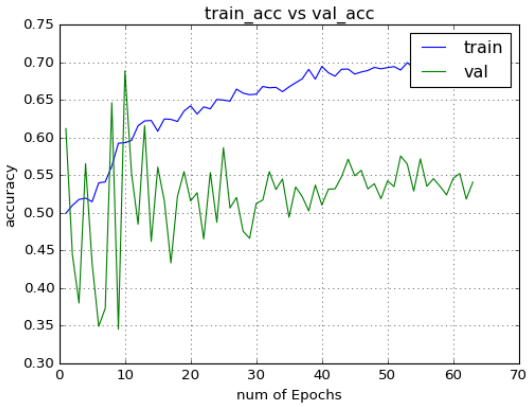
CNN for NATIVE acc=0.55%

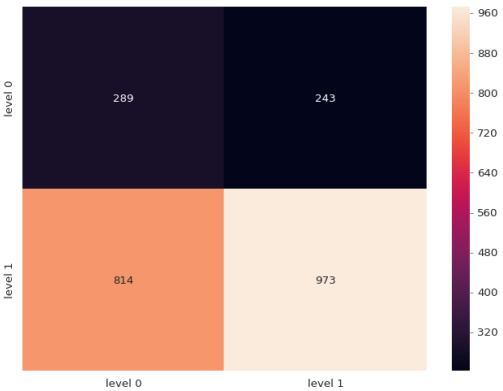


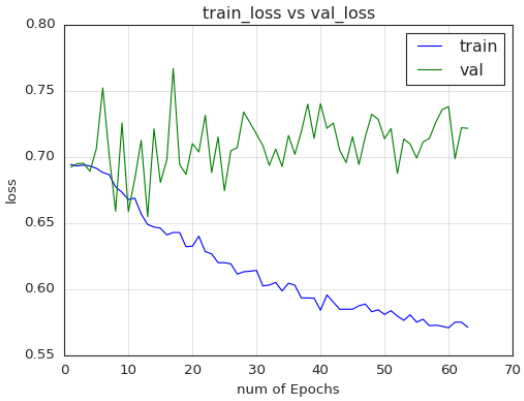



```
predicted variable: native

Test Loss: 0.85547935962677
Test accuracy: 0.5532557368278503
73/73 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

     level 0       0.25      0.49      0.33       532
     level 1       0.79      0.57      0.66      1787

    accuracy                           0.55      2319
   macro avg       0.52      0.53      0.50      2319
weighted avg       0.67      0.55      0.59      2319

confusion matrix:
 [[ 259  273]
 [ 763 1024]]
Balanced acc score: 0.5299347627602863
Balanced error rate: 0.4700652372397137
```

RNN for NATIVE acc=0.54%








```
predicted variable: native

Test Loss: 0.7707700729370117
Test accuracy: 0.54420006275177
73/73 [==============================] - 2s 8ms/step
              precision    recall  f1-score   support

     level 0       0.26      0.54      0.35       532
     level 1       0.80      0.54      0.65      1787

    accuracy                           0.54      2319
   macro avg       0.53      0.54      0.50      2319
weighted avg       0.68      0.54      0.58      2319

confusion matrix:
 [[289 243]
 [814 973]]
Balanced acc score: 0.5438605256846649
Balanced error rate: 0.4561394743153351
```

Both CNN and RNN perform poorly below Null Accuracy

The model is built to study and analyze approaches to classify a reader's comprehension level. It worked on two prediction variables: the reader's difficulty and whether the reader is a native speaker of English. The two deep network models, CNN and RNN, inputted with windows of 21 consecutive fixations of an SAT passage, tried to predict the accuracy of the prediction variables. It is observed that prediction on new passages for native variables gives better performance when compared to difficulty variables. Prediction on new reading windows for native is better than for difficulty. The CNN model worked well in most cases compared to the RNN model. Predictions on new readers for difficulty and native are very poor. There is still a lot of advancement to be done for the model. Thus, predicting comprehension from eye-movement data is a problem that needs to be worked upon. The model shows the difference in statistical testing and prediction. Though the model could not perform well, it still predicted accuracies with some success rate. Thus, these models are being used in various fields, and every improvement in the predictions can make a significant difference in diverse contexts.

DISCUSSION

From the model, it is known that there is a lot to improve in the model's working. The models gave good accuracies in some cases, while they performed poorly in others. It is seen that the performance of the models is better when it comes to the native variable, while it is a bit bad in the case of the difficulty variable for new reading windows and new passages. They performed equally poorly in the case of new readers. It is also observed that the CNN model works better than the RNN model in many of the points.

The major limitation of this approach is the assumption that ground truth labels are assigned to local fixation windows based on the passage-level labels. This is done to avoid the task of labeling manually or one by one during compilation. But this problem needs to be solved for better performance of the models. A feasible solution to solve this problem can be to learn a weighting of local features that optimizes global prediction. This can be done by computer vision. A study is already going on on this issue to construct reading detectors without using explicit ground truth labels for local fixation windows. It is an assumption that the inclusion of lexical features could also improve the performance of the models.

It is also observed that the results generated from the models vary as the dataset is randomly created. The training, validation, and testing datasets for the same preliminary dataset differed; the resulting accuracies varied. Thus, the accuracies listed here may not be optimal, but with few increases or decreases from the optimal accuracies. Most of cases, the accuracies obtained are below the null accuracy. Still, these models' performance needs to be improved and is to be worked upon. Thus, it concludes the working and implementation of the model for predicting reading comprehension from gaze behavior.

REFERENCES

Hosam Al-Samarraie, Atef Eldenfria, and Husameddin Dawoud. 2017. The impact of personality traits on users' information-seeking behavior. Information Processing & Management 53, 1 (2017), 237–247.

Hosam Al-Samarraie, Samer Muthana Sarsam, Ahmed Ibrahim Alzahrani, and Nasser Alalwan. 2018. Personality and individual differences: the potential of using preferences for visual stimuli to predict the Big Five traits. Cognition, Technology & Work (2018), 1–13.

Olivier Augereau, Hiroki Fujiyoshi, and Koichi Kise. 2016. Towards an automated estimation of English skill via TOEIC score based on reading analysis. In 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 1285–1290.

Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust real-time reading-skimming classifier. In Proceedings of the Symposium on Eye Tracking Research and Applications. ACM, 123–130.

Jonathan FG Boisvert and Neil DB Bruce. 2016. Predicting task from eye movements: On the importance of spatial distribution, dynamics, and image features. Neurocomputing 207 (2016), 653–668.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18, 5-6 (2005), 602–610.

John M Henderson. 2003. Human gaze control during real-world scene perception. Trends in cognitive sciences 7, 11 (2003), 498–504.

John M Henderson, Svetlana V Shinkareva, Jing Wang, Steven G Luke, and Jenn Olejarczyk. 2013. Predicting cognitive state from eye movements. PloS one 8, 5 (2013), e64937.

Shoya Ishimaru, Kensuke Hoshika, Kai Kunze, Koichi Kise, and Andreas Dengel. 2017. Towards reading trackers in the wild: detecting reading activities by EOG glasses and deep neural networks. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 704–711.

Conor Kelton, Zijun Wei, Seoyoung Ahn, Aruna Balasubramanian, Samir Das, Dimitris Samaras, and Gregory Zelinsky. 2019. Reading Detection in Real-time. In In 2019 Symposium on Eye Tracking Research and Applications (ETRA '19).

Ya Lou, Yanping Liu, Johanna K Kaakinen, and Xingshan Li. 2017. Using support vector machines to identify literacy skills: Evidence from eye movements. Behavior research methods 49, 3 (2017), 887–895.

Silvia Makowski, Lena A Jäger, Ahmed Abdelwahab, Niels Landwehr, and Tobias Scheffer. 2018. A Discriminative Model for Identifying Readers and Assessing Text Comprehension from Eye Movements. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 209–225.

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

Keith Rayner. 2009. Eye movements and attention in reading, scene perception, and visual search. The quarterly journal of experimental psychology 62, 8 (2009), 1457–1506.

Geoffrey Underwood, Alison Hubbard, and Howard Wilkinson. 1990. Eye fixations predict reading comprehension: The relationships between reading skill, reading speed, and visual inspection. Language and speech 33, 1 (1990), 69–81.

Zijun Wei, Hossein Adeli, Minh Hoai Nguyen, Greg Zelinsky, and Dimitris Samaras. 2016. Learned Region Sparsity and Diversity Also Predicts Visual Attention. In Advances in Neural Information Processing Systems. 1894–1902.