Reading Type Classification based on Generative Models and Bidirectional Long Short-Term Memory

Seyyed Saleh Mozaffari Chanijani
TU Kaiserslautern
German Research Center for Artificial Intelligence
mozafari@dfki.uni-kl.de

Federico Raue
TU Kaiserslautern
German Research Center for Artificial Intelligence
Federico.Raue@dfki.de

Saeid Dashti Hassanzadeh
TU Clausthal
saeid.dashti.hassanzadeh@tu-clausthal.de

Stefan Agne
German Research Center for Artificial Intelligence
Stefan.Agne@dfki.de

Syed Saqib Bukhari
German Research Center for Artificial Intelligence
Saqib.Bukhari@dfki.de

Andreas Dengel
TU Kaiserslautern
German Research Center for Artificial Intelligence
dengel@dfki.uni-kl.de

ABSTRACT

Measuring the attention of users is necessary to design smart Human Computer Interaction (HCI) systems. Particularly, in reading, the reading types, so-called reading, skimming, and scanning are signs to express the degree of attentiveness. Eye movements are informative spatiotemporal data to measure quality of reading. Eye tracking technology is the tool to record eye movements. Even though there is increasing usage of eye trackers in research and especially in psycholinguistics, collecting appropriate task-specific eye movements data is expensive and time consuming. Moreover, machine learning tools like Recurrent Neural Networks need large enough samples to be trained. Hence, designing a generative model in order to have reliable research-oriented synthetic eye movements is desirable. This paper has two main contributions. First, a generative model in order to synthesize reading, skimming, and scanning in reading is developed. Second, in order to evaluate the generative model, a bidirectional Long Short-Term Memory (BLSTM) is proposed. It was trained with synthetic data and tested with real-world eye movements to classify reading, skimming, and scanning where more than 95% classification accuracy is achieved.

ACM Classification Keywords

I.5.4. Computing Methodologies: PATTERN RECOGNITION; Applications;

UISTDA '18, March 11, 2018, Tokyo, Japan

Author Keywords

Eye Tracking; reading type; classification; synthetic data; generative models; Hierarchical Hidden Markov Models; Gaussian Mixture Models; LSTM; Recurrent Neural Networks; reading; skimming; scanning;

INTRODUCTION

Reading, is the ability to extract visual information from the page and comprehend the meaning of underlying text [16]. Considering attention, as presented in Figure 1, the reading types is divided into three categories: reading, skimming, and scanning. On eye tracking context, the reading is a method of moving the eyes over the text to comprehend the meaning of it. The skimming is a rapid eye movement over the document with the purpose of getting only the main ideas and a general overview of the document whereas scanning rapidly covers a lot of contexts in order to locate specific fact or piece of information.

The fixation progress on words expressed in character units must be measured in order to detect the reading type, i.e., deciding whether observed eye movement patterns in the reading types [4, 12]. This approach applies in cases where the eye tracking accuracy is high enough to provide word level resolution [16]. Example applications include ScentHighlight [6], which highlights related sentences during reading; the eyeBook [2], where ambient effects are to be triggered in proximity of the reading position; or QuickSkim [3], where non-content words may be faded out in real time with an increase of skimming speed to make reading more efficient.

Due to the noisy nature of the eye tracking apparatus where the point of gaze cannot be determined exactly, it can be desirable to automatically decide to what extent eye movements resemble a reading type patterns. In this regard, a psycholinguist is able to determine what segments of the scanpath belong to reading, skimming, or scanning, even though the fixations do not match the underlying text.

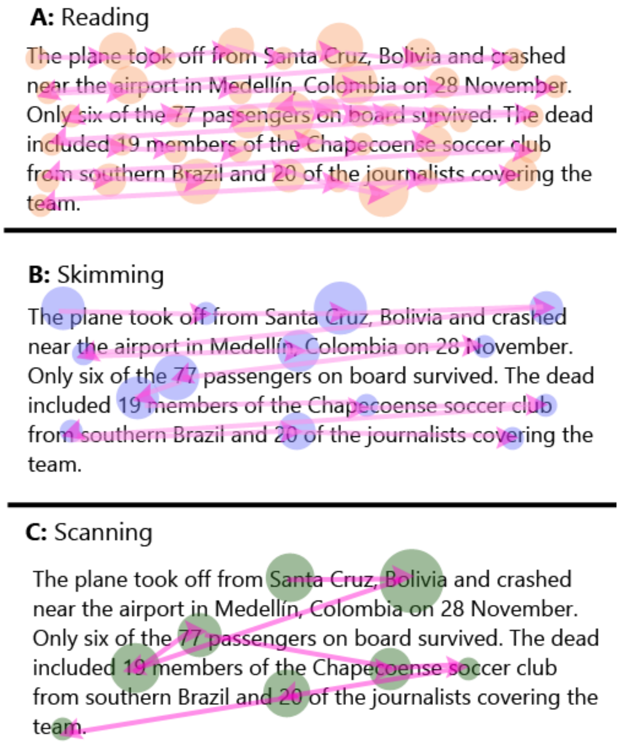

Figure 1: The three reading types. A: Reading; the saccades are short and progressive over the context. B: Skimming; here saccades are longer compare to the reading pattern. C: Scanning; compare to skimming the saccades are less unstructured.

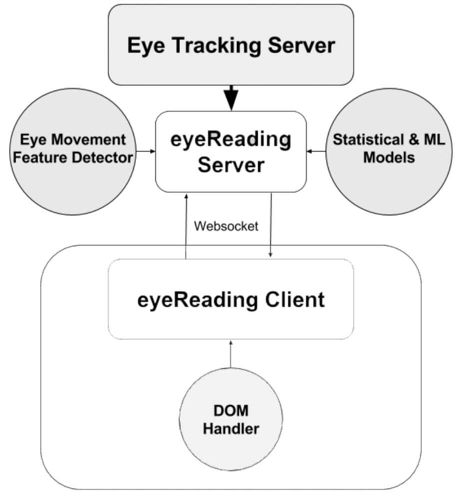

Figure 2: The top-level architecture of the eyeReading framework. The eyeReading Server and eyeReading Client components are connected via WebSocket protocol. [14]

Biedert et al. [4] proposed a reading-skimming classifier. Despite the model classifies the reading and the skimming patterns, it does not cover scanning patterns. It is desirable to have scanning patterns in order to have better estimation on the degree of attiontion in reading to designing a proper Human Document Interaction system. In addition, in the domain of information retrieval, it has been shown that acquiring implicit feedback from a reading type detection, including scanning, can significantly improve search accuracy through personalization [5].

Detecting of the reading types is a *sequence classification* problem. The term sequence classification encompasses all tasks where sequences of data are transcribed with sequences of discrete labels [9]. The discrete labels in the reading type classification problem are shown in Figure 1. Long Short-Term Memory (LSTM) [10] is a variation of Recurrent Neural Networks (RNNs) which is suitable to classify the sequential data. However, such networks need large enough samples for training. Unfortunately, the task-specific eye tracking data size would not big enough to be applied in the deep networks. Hence, there is a need to synthesize task-specific eye movement patterns in order to deploy deep neural networks.

In this paper, we propose a generative model which synthesizes the reading types patterns. We also designed and evaluated a BLSTM model. The model trained on both the original dataset as well as the synthetic dataset.

The paper is structured as follows. We start with presenting an experiment conducted to collect real-world eye movements in reading in order to build a reference for data synthesization and the reading type classification. Then, a two-layered Hierarchical Hidden Markov Model (HHMM) for eye movement data synthesization is proposed. Moreover, we present a BLSTM-based sequential model to detect and classify reading, skimming, and scanning. This model built on features described in section Features and Training. In the Evaluation and Result section, we evaluate our models and describe our results. This is followed by our conclusion.

REAL-WORLD DATA ACQUISITION

The first step of constructing a system which is able to learn and distinguish eye movement patterns of reading types is to record the real-world eye movement data during reading mode. The recorded data must comprise all possible state categories: reading, skimming, and scanning. To perform the task, we designed an experiment to record eye movements of ten participants from the local university. We used this data as a reference to build up a Hierarchical Hidden Markov Model (HHMM) for synthetic data generation. Furthermore, this real-world data partially employed for testing and evaluating the classifiers.

Apparatus

In this study, we deploy SensoMotoric Instrument iViewX scientific REDn eye tracker operating at 60Hz. The tracking error reported by the manufacturer was less than 0.4 degree,

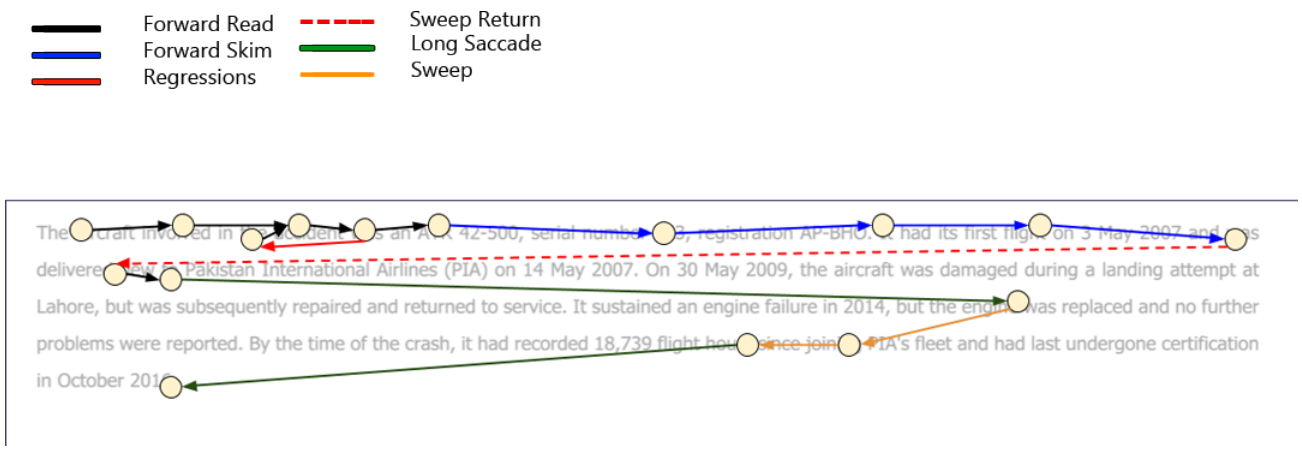

Figure 3: The feature components of our study: The *fixations* are shown by yellow circles. A *regression* is an implicit sign that the reader is having difficulty understanding the material. It is shown by the red gaze path. When processing the fixations to form forward reads, a forward read will be stopped when (1) a regression is encountered, (2) a forward saccade is too large and likely a forward skip through the text, or (3) the eye gaze moves to another line of text. The last called *sweep return* and indicated with dashed red color in the figure. [16] [13].

which makes it appropriate for fixation-based eye movement studies. [1]

Experimental Setup

The experiment was designed in which all the three reading types could be obtained. Two articles in plain English were chosen from Wikipedia. They are about two airplane crashes took place in Colombia[2] and Pakistan[3] in 2016. Ten participants from local university participated in our study. None of them were native Engish speaker but they were fluent in English as the second language. In the first phase of the experiment, the participants were requested to write a comprehensive report on the article they had given to read. Hence, they would read the selected article thoroughly. In the second phase, the participants were asked to find the specific information in the second article, e.g., how many crews were in the airplane or what was the flight number. Therefore, most of the eye movement patterns associated with skimming and scanning. Consequently, all three reading type patterns recorded during trials. The trials have been recorded in specialized eye-tracking interface *eyeReading* [14]. The eyeReading facilitates research in reading psychology and provides a framework for gaze-based Human Document Interaction. Figure 2 shows top-level architecture of eyeReading.

[1]https://www.smivision.com/eye-tracking/product/redn-scientific-eye-tracker/
[2]https://en.wikipedia.org/wiki/LaMia_Flight_2933
[3]http://en.wikipedia.org/wiki/Pakistan_International_Airlines_Flight_661

Figure 4: The application designed for two steps annotation. A sequence buffer of window size 5 presented to the annotator. This made to facilitate annotator's decision for the state label as well as the saccade label of the red saccade. The orange circles are the fixation durations and the bigger size implies the bigger duration.

FEATURES AND ANNOTATION

The recorded raw gaze information must be processed in order to extract saccadic features associated with reading. The extracted saccadic features are the length of the saccade, velocity of the saccadic movement, fixation duration associated with the saccades, and angularity of the saccade. In this section, first, we demonstrate the features extraction step and then the process of making ground-truth will be explained.

Features

On account of inevitable noise in the eye tracking trials, we first applied a virtual median filter on the input raw gaze points $E' = e'_1, ..., e'_n$ to eliminate any possible outliers.

$$e_i = (med_x(e'_{i-2}, ..., e'_{i+2}), med_y(e'_{i-2}, ..., e'_{i+2})) \qquad (1)$$

In the second step, the fixations were detected using dispersion method [11]. We considered 100*ms* temporal and 50*px* spatial for dispersion parameters. The saccades are considered as two consecutive fixations with the following features:

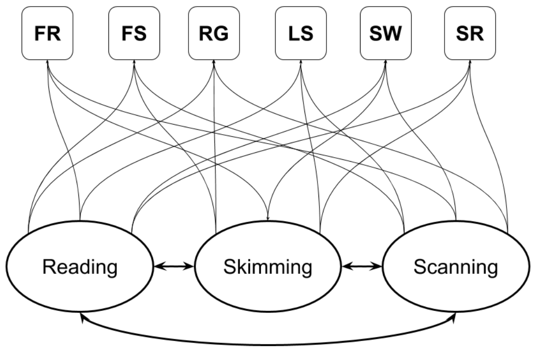

Figure 5: The first layer of the probabilistic model for the reading types data generation. An HMM where the states are our class labels and the emissions are the saccade labels. The output is the sequence of class labels with the corresponding length.

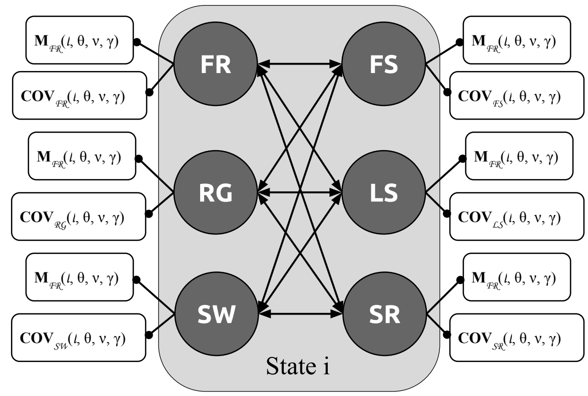

Figure 6: The second layer of the probabilistic model for the reading types data generation. Here, in the GMM-HMM, the states are the saccades' labels and the components (features) value calculated through Gaussian distribution with respect to the covariance matrix and the mean matrix of the feature set $F : \ell, \theta, \nu, \gamma$.

1. **amplitude**(ℓ): the distance between two progressive fixations in virtual character unit (vc).

2. **angularity**(θ): the angle of the saccade respect to its starting point. The α indicates the direction of the saccade in circle domain: $-180° \leq \alpha \leq 179°$.

3. **velocity**(ν): the speed of the saccade:

$$\nu = \frac{\ell}{t_e - t_s} \quad (2)$$

where ts and te are the first timestamps in the start fixation and the end fixation of a saccade in milliseconds.

4. **duration**(γ): The start fixation duration in each saccade in milliseconds respectively.

Therefore, $F(\ell, \theta, \nu, \gamma)$ are the features selected for the saccades in the collected data.

Data Annotation

After the feature extraction step, we designed a labeling application to make ground truth from data. The figure 4 shows the interface of the labeling application. Ground-truthing data was accomplished in two steps: the saccade labeling and the sequence labeling.

Saccade Labeling

In the first step of labeling, the expert made a judgment about the saccades with respect to the provided features $F : \ell, \alpha, \nu, \theta$.The saccades grouped into six categories or labels [13].

- FR: Forward Read (FR) is a progressive saccade associated with reading. The amplitude is between 7 to 10 characters [16].
- FS: Forward Skim(FS)is also a progressive saccade which the amplitude is larger than FR but not too large.
- *LS*: Long Saccades(*LS*) is those saccades which have the bigger amplitude than the threshold considered for the context. The direction of saccade does not apply to *LS*.
- *RG*: The regressions(*RG*) are regressive saccades usually associated with reading which is the sign of difficulties in reading. The amplitude is varied and it must target the passed context.
- *SR*: Sweeping back to the left on the next line of text is called the Sweep Returns (*SR*).
- SW: Unstructured sweeping the text to look up information are labeled as Sweeps (SW).

Figure 1 intuitively shows the difference between the six saccade labels.

Sequence Annotation

After all the saccades were labeled in the previous section into the six categories, in the second step, the sequences made by these saccades annotated as reading, skimming, or scanning. At the end, 396 annotated sequences for reading, 378 for skimming, and 118 for scanning were acquired.

SYNTHETIC EYE MOVEMENTS IN READING

It is always desirable to have enough data samples to construct robust machine learning models. Especially, in deep neural networks, a very big training set is usually required. Unfortunately, appropriate data acquisition in eye tracking studies is

Algorithm 1: Algorithm to simulate a HMM states sequence S given the model $\lambda = \{\Pi, A, B\}$.

Data: $states_{ground-truth} = s_1, s_2, ..., s_n$ and $observation_{ground-truth} = o_1, o_2, ..., o_n$ where n is the number of saccades in the ground-truth.

Result: States sequence $S_{sequence} = ([s_1, l_1], [s_2, l_2], ..., [s_k, l_k])$ where k is number of sequences, s is the sequence label $s_i \in C$, and l_i is the length of s_i.

$\Pi = (0.34, 0.33, 0.33)$;
$A = \{a_{ij} | i = 1,2,3; j = 1,2,3\}$: State transition probability where $a_{ij} = P(s_{t+1} = j | s_t = i)$ and $\sum_{j=1}^{3} a_{ij} = 1$;
$B = \{b_k(o_t) | k = 1,2,3; t = 1,...,n\}$: Observation probability where $b_k(o_t) = P(o_t | s_t = k)$;
Choose an initial state S_1 according to the initial state distribution π;
for *time* $t = \{1,...,n\}$ **do**
  Draw o_t from the probability distribution $B_s t$;
  Go to state $s_t + 1$ according to the transition probabilities A_{s_t};
  Set $t = t + 1$;

Algorithm 2: Algorithm to generate the emissions sequence S' from S.

Data: S and $observation_{ground-truth} = o_1, o_2, ..., o_n$ where n is the number of saccades in the ground-truth data.

Result: synthetic emissions for all $s \in S$

$\Pi' = (\pi'_1, ..., \pi'_6)$: Initial state probabilities where $\pi'_i = P(s'_1 = i)$ and $\sum_{i=1}^{6} \pi'_i = 1$;
$A' = \{a'_{ij} | i = 1,...,6; j = 1,...,6\}$: Emissions transition probability where $a'_{ij} = P(s'_{t+1} = j | s'_t = i)$ and $\sum_{j=1}^{6} a'_{ij} = 1$;
foreach states *reading*, *skimming*, *scanning* calculate the covariance matrix COV of the emissions FR, FS, RG, LS, SW, SR for the features $\ell, \theta, \nu, \gamma$;
foreach states *reading*, *skimming*, *scanning* calculate the mean matrix M of the emissions FR, FS, RG, LS, SW, SR for the features $\ell, \theta, \nu, \gamma$;
Choose an initial state S'_1 according to the initial state distribution Π';
for *time* $t = \{1,...,n\}$ **do**
  Draw o_t from the Gaussian distribution with cov_{st} and $mean_{st}$;
  Go to state $s_t + 1$ according to the transition probabilities A'_{s_t};
  Set $t = t + 1$;

not an easy task. It needs eye trackers which are still expensive in the market as well as an appropriate experimental setup designed for a specific goal, i.e., reading type classification. Hence, the idea is to generate task-specific eye movement data from the real-world data in which these synthetic data can be used to construct a better model and even to use in other applications and research. It motivated us to design a *Hierarchical Hidden Markov Model* (HHMM) to generate synthetic eye movements in reading. Usually, ordinary Markov chains are often not flexible enough for the analysis of real-world data, as the state corresponding to a specific event (observation) has to be known. However, in many problems of interest, this is not given. Hidden Markov models (HMM) as originally proposed by Baum et al. (1970) [1] can be viewed as an extension of Markov chains. The only difference compared to common Markov chains is, that the state sequence corresponding to a particular observation sequence, i.e., reading types in our case, is not observable but hidden. In other words, the observation is a probabilistic function of the state, whereas the underlying state sequence itself is a hidden stochastic process [15]. That means the underlying state sequence can only be observed indirectly through another stochastic process that emits an observable output. Hidden Markov models are extremely popular when dealing with sequential data, such as speech recognition, character recognition, gesture recognition as well as biological sequences. Therefore, the HMM is a right candidate to handle the eye movement patterns where they are sequential and by nature stochastic. In order to synthesize our data, the graphical model should be able to generate both saccadic sequences and reading state sequences. Therefore, in this paper, a two-layered Hierarchical HMM is designed. In an HHMM each state is considered to be a self-contained probabilistic model [8]. Briefly, in the first layer as shown in Figure 5 we modeled the reading, skimming, and scanning as states of the Markov model and emissions are FR (Forward Reading), FS (Forward Skimming), SR (Sweep Return), RG (Regression), SW (Sweeps), LS (Long Saccades). As shown in Figure 6, each of states in the first level is self-contained mixture graphical model so-called GMM-HMM (Hidden Markov Model with Gaussian Mixture Model). This layer responsible to generate values for the four mentioned saccadic features $F : \ell, \alpha, \nu, \theta$.

Method

The 2-layered HHMM constructs the probabilistic model that generates saccades associated with the reading types. It is a top-down approach to synthesize natural reading types. The task of the first layer, which is shown in Figure 5, is to generate

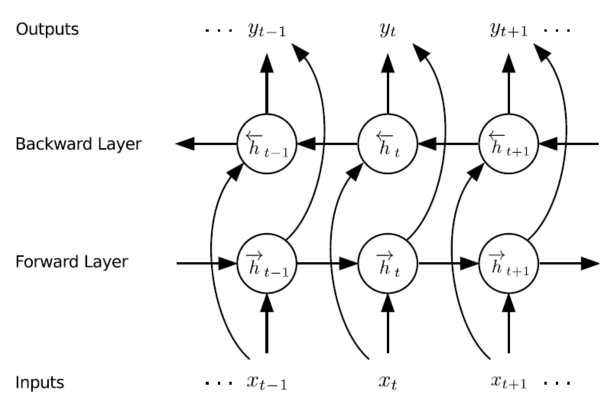

Figure 7: BLSTM architecture: the forward (resp. backward) layer processes the input sequence in the (resp. reverse) order. Output layer concatenates hidden layers values at each timestep to make a decision by considering both past and future contexts [9].

N	Precision	Recall	Accuracy
5	0.81	0.79	0.784
8	0.89	0.90	0.896
10	0.93	0.93	0.925

Table 1: The results of BLSTM model on the original dataset. 196 testing sequences distributed in 75 for reading, 83 for skimming, and 38 scanning.

the sequence of the states reading, skimming, and scanning. Algorithm 1 constructs this layer. In order to build the 1^{st} layer HMM, $\lambda(\pi, A, B)$, the transition state matrix A and emission matrix B are built upon the labeled data explained in section *Data Labeling*. We considered equal probabilities (33%) for the states of reading types in π. Then, the states *reading*, *skimming*, and *scanning* is generated based on multinomial distribution.

The algorithm 2 presents the construction of the second layer of our graphical model so-called GMM-HMM model. Where the input is the state sequences produced from the first layer. In contrast to the first layer, the emissions (observations) associated with each sequence are generated based on Gaussian distribution. Hence, for each state (reading, skimming, and scanning), it needs to compute the transition matrix of observation, the mean of each component (features $F : \ell, \theta, \nu, \gamma$) as well as the covariance matrix of the features. Figure 6 presents the second layer of the model.

READING TYPE CLASSIFICATION WITH BLSTM

Recurrent neural networks (RNNs) are able to access a wide range of context and sequences [17]. However, standard RNNs make use of the previous context only whereas bidirectional RNNs (BRNNs) are able to incorporate emissions on both sides of every position in the input sequence [18]. This is useful in the problem of reading type detection since it is often necessary to look at both sides to the right and to the left of a given sequence in order to identify it. BLSTM is a BRNN that has hidden layers, which are made up of the so-called Long Short-Term Memory (LSTM) cells. LSTM is an RNN architecture specifically designed to bridge long time delays between relevant input and target events, making it suitable for problems where long-range emission sequences are required to disambiguate individual labels [10]. In fact, BLSTM networks suit well for the reading type detection. Figure 7 shows BLSTM-RNN architecture and
Figure 8 presents the model implemented for our study with keras [7]. It consists of two BLSTMs, one dropout to prevent overfitting, and two dense networks. Where n is the sequence length of the input. The loss function is *categorical crossentropy* and *SoftMax* is used as the activation function.

EVALUATION AND RESULTS

In this section, the evaluation of both generative model and the classifier is presented. Here, two types of the data are available; the original data recorded with the eye tracker (actual data) and the synthetic data. The actual data was randomly split into the train set 60%, validation set 10%, and test set 30%.First half of the actual data was used in the generative model for data synthesization. The other half used for testing. In all cases, the train set first fitted with standard scale function to scale the mean (μ) to 0 and the standard deviation(δ) to 1. Then the validation and test set transformed respect to fitted data.
The model trained with different sequence length $N = 5, 8, 10$.

Baseline: Actual data and SVM-RBF classifier

The baseline is to test and evaluate SVM-RBF classifier proposed by Biedert et al. [4]. For the test set, there are only 196 sequences to support the model. 75 for reading, 83 for skimming, and 38 scanning. By 5 cross-fold validation the best accuracy acquired was 69% with parameters $C = 1000$ and $gamma = 0.001$. With a closer look at the confusion matrix in figure 9, it is obvious that there is an unacceptable confusion in the class *scanning*. This problem is on account of the less number of supports for the class *scanning* where there is just 19% of the class labels. Another reason is about sequential characteristics of the data. It shows Support Vector Machines are not the best machine learning model tailored for such data.

Proposed method: Actual data and BLSTM

The model presented in Figure 8 is used to train and test the original data. The model trained with different sequence length $N = 5, 8, 10$. In case, the input sequence has a different length, the sequence padded or truncated to the fixed length n. Table 1 show the accuracies for the different length. 92.5% accuracy achieved for sequence length of 10.

roposed method: Synthetic data and BLSTM

Finally, the BLSTM model trained with synthetic data which was generated with the half of the original dataset. The same half of the original dataset also was used to validating the model. The second half of the original dataset used for testing. Table 2 presents the results for different variation of data size and sequence length. While the larger sequence windows has the better results in the model, an instant user interface favors smaller sequences. The length of a sequence is related to the number of fixations. If we consider the average fixation duration in reading $250ms$ [16], the model must wait for the input sequence for $2.5s$. This result supports the reliability of the synthetic data. The larger data, the better performance as expected in deep net frameworks.

CONCLUSION

Recurrent Neural Networks (RNNs) are suitable to model spatiotemporal data, i.e, eye movements in reading. They usually need large enough data sample to be trained. On account of constraints in the experimental setups, accessibility to the expensive eye tracking apparatus, and finding appropriate and enough participants, there is usually lack of enough data in eye tracking research to employ RNNs. In this paper a novel probabilistic approach for eye movements data synthesization in reading is proposed. Also a BLSTM model for both the original recorded data and the synthetic data in order to classify the reading types: reading, skimming, and scanning is presented. The RNN-based classifier proposed in this paper achieved more than 95% accuracy in the reading type detection which not only outperforms the previous works but also

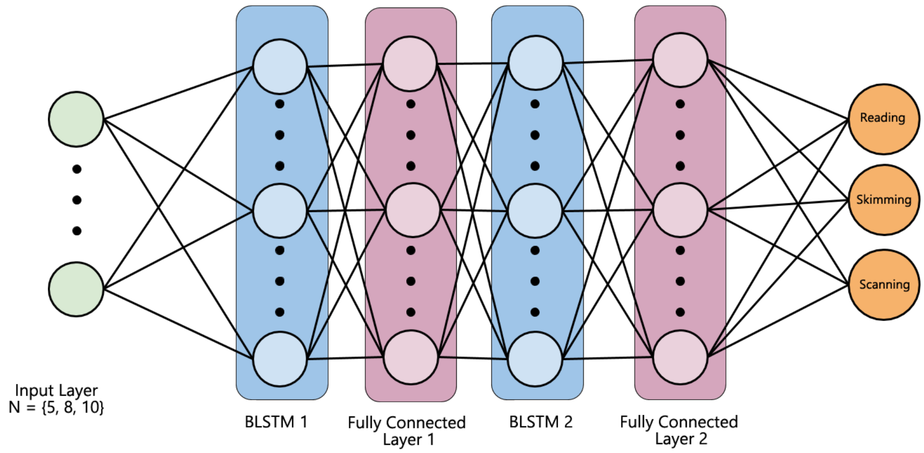

Figure 8: The BLSTM model used in our study. It consist of two BLSTMs, one dropout to prevent overfitting, and a two fully connected networks. Here, N is the sequence length of the input. The loss function was *categorical cross-entropy* and *Softmax* is used as the activation function.

Synthetic Data Size	N	Precision	Recall	Accuracy
5K	5	0.87	0.86	0.86
5K	8	0.92	0.91	0.91
5K	10	0.93	0.93	0.93
10K	5	0.88	0.87	0.87
10K	8	0.92	0.92	0.92 *
10K	10	0.94	0.93	0.94
50K	5	0.88	0.92	0.92
50K	8	0.95	0.94	0.94
50K	10	0.96	0.95	0.95 *

Table 2: The result of BLSTM model trained with synthesize data. The result supports the reliability of the synthetic data. The larger data, the better performance. The larger sequences show the better results but a realtime classifier favors the shorter sequences.

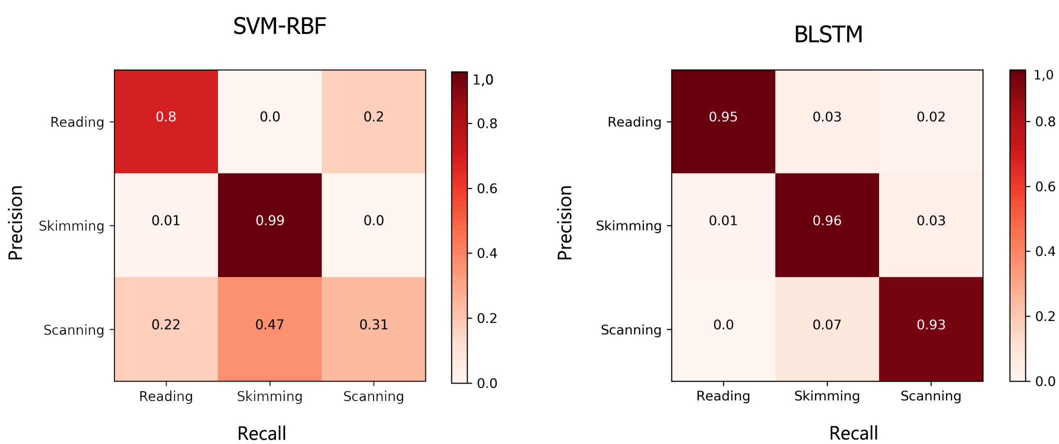

Figure 9: Confusion matrix for two base lines. The left confusion matrix shows high confusion between scanning and skimming whereas BLSTM confusion matrix (the synthetic data with size of 10^5 with sequence length $N = 5$) shows robustness of the model.

contains the *scanning* reading type.
One important note is on the sterategy on selecting the sequence length (N) for the model. Even though the longer sequence length would lead to the higher accuracy, in order to design instant user interface the shorter length is more desirable as the model waits for N number of saccades to classify. Depend on the application, the sequence length could be selected occasionally.
The outcome of this research is promising in which appropriate data synthesization breaks limitations on using RNNs in eye tracking research in general. It also may offers the possibilities to provide standard eye movement datasets not only for the reading type detection but for other research aspects in reading i.e., in the research about dyslexia. Also, it gives insight to other eye-tracking research to generate eye movement transitions in different area of interests which is very helpful to distinguish experts and novices in several domains of education. It is also desirable to explore alternative to HMM for data synthesization. In this regard, using LSTM itself as a generative model for eye tracking data is in our agenda of research for the future work.

Acknowledgment
This work was funded by the Federal Ministry of Education and Research(BMBF) for the project AICASys.

REFERENCES

1. Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics* 41, 1 (1970), 164–171.
2. Ralf Biedert, Georg Buscher, and Andreas Dengel. 2010a. The eyebook–using eye tracking to enhance the reading experience. *Informatik-Spektrum* 33, 3 (2010), 272–281.
3. Ralf Biedert, Georg Buscher, Sven Schwarz, Jörn Hees, and Andreas Dengel. 2010b. Text 2.0. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4003–4008.
4. Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 123–130.
5. 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 (TiiS)* 1, 2 (2012), 9.
6. Ed H Chi, Lichan Hong, Michelle Gumbrecht, and Stuart K Card. 2005. ScentHighlights: highlighting conceptually-related sentences during reading. In *Proceedings of the 10th international conference on Intelligent user interfaces*. ACM, 272–274.
7. François Chollet. 2015. keras. `https://github.com/fchollet/keras`. (2015).
8. Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden Markov model: Analysis and applications. *Machine learning* 32, 1 (1998), 41–62.
9. Alex Graves and others. 2012. *Supervised sequence labelling with recurrent neural networks*. Vol. 385. Springer.
10. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
11. Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. 2011. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford.
12. Aulikki Hyrskykari. 2006. *Eyes in attentive interfaces: Experiences from creating iDict, a gaze-aware reading aid.* Tampereen yliopisto.
13. Seyyed Saleh Mozaffari Chanijani, Mohammad Al-Naser, Syed Saqib Bukhari, Damian Borth, Shanley EM Alleny, and Andreas Denge. 2016a. An eye movement study on scientific papers using wearable eye tracking technology. In *Mobile Computing and Ubiquitous Networking (ICMU), 2016 Ninth International Conference on*. IEEE.
14. Seyyed Saleh Mozaffari Chanijani, Syed Saqib Bukhari, and Andreas Dengel. 2016b. eyeReading: Interaction with Text through Eyes. In *Proceedings of The Ninth International Conference on Mobile Computing and Ubiquitous Networking*, Vol. 2016. 1–2.
15. Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.
16. Keith Rayner, Alexander Pollatsek, Jane Ashby, and Charles Clifton Jr. 2012. *Psychology of reading*. Psychology Press.
17. Raúl Rojas. 2013. *Neural networks: a systematic introduction*. Springer Science & Business Media.
18. Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.