Automatically Detecting Reading in Eye Tracking Data

Sepp Kollmorgen and Kenneth Holmqvist

June 8, 2007

Abstract

To date, more and more eye tracking based studies are conducted to investigate reading under natural conditions (e.g. long texts, texts embedded with pictures etc.) and to investigate the perception of every day stimuli containing text (e.g. web pages, newspapers). When the data of such studies are to be statistically analyzed, it is often necessary to know when subjects are actually reading. Assuming that subjects are reading whenever they look at text is often a poor solution, because it would, for example, include scanning which is a very different process. In this paper, we depict a new, more elaborate but concise method to detect reading based on modeling fixation-saccade sequences with a Hidden Markov Model. The model can be used both off-line and on-line.

1 Introduction

To date, more and more eye tracking based studies are conducted to investigate reading under natural conditions and to investigate the perception of every day stimuli containing text [1]. Such stimuli could be web pages, newspapers or textbooks. In most cases, such stimuli do not only contain text but also pictures and their viewing falls into different types of processes, such as reading and picture viewing. To distinguish these processes is important when it comes to statistical analysis of experimental data. A widely used method for distinction is Region of Interest Analysis (ROI): it is assumed that a subject is reading when she looks at text. Such a simple analysis may fail in situations where looking at a text does not always mean reading it. Consider, for example, the case of a writer who monitors and edits her own text: during editing periods her gaze may rest on text ROIs, allthough she is not reading. In these cases, a more sophisticated technique is needed. We suggest to exploit the fact that certain eye movement patterns are characteristic for reading and use a Hidden Markov Model, which learns the key properties of these patterns, to recognize them. Our approach uses tracked eye movements only, it is independent of ROIs or any other method which takes the stimuli into consideration.

We tested our approach on a large amount of experimental data which have been recorded using different stimuli. We labelled instances of reading and non-reading in a fraction of that data to obtain a training and a validation set. Section 2 describes the experimental data and the formation of the training and the validation set. The Hidden Markov Model and the learning algorithms are depicted in section 3, section 4 sketches how it can be used on-line and section 5 compares it to a neural network approach. The final section 6 evaluates and discusses the approach.

2 The Data

The data we used for training and evaluation have been recorded in the course of an experiment investigating text production. The subjects were instructed to write a text either describing a picture or discussing a film they had just seen. They used a computer to write. During the writing phase, their gaze positions on the monitor (in a 1024x768 coordinate system) and on the picture were recorded[1]. Addition-

[1] An SMI iView X head mounted eye tracking system with Polhemus head tracking has been used

ally, videos with the field of view and the location of gaze were recorded. The experiment had 96 participants, about a third of them had reading and writing difficulties [2].

To devise the training and evaluation data we labelled instances of reading and non-reading in 20 randomly chosen datasets and dissected this group into two parts, a training and an evaluation set. All in all, we had about about 1 hour of labelled data. The decision of whether a certain gaze-sequence reflects reading or not was based on its appearance and on the respective video material. We labelled only clear cases of reading and non-reading and omitted dubious ones. Furthermore, we only labeled instances that encompassed at least three fixations and made no difference between reading consisting mainly of forward and reading consisting mainly of backward saccades. This working definition of reading in our opinion generally coheres to what is considered to be the characteristics of reading as described by Rayner (1998) and others [3].

2.1 Preprocessing

After the eye tracking recordings are done the data of one subject are available as a list of gaze points. In our case, we have one gaze point for each 20 milliseconds of the recording period. Depending on the experimental paradigm, it may be sensible to define an ROI (Region of Interest) and to dismiss gaze points that fall out of it. In our case, only the computer monitor was of interest regarding the detection of reading. This preselection gives one or several sequences of gaze points all lying in the specified ROI.

As not the gaze point sequences themselves are of interest, but the sequences of saccades and fixations corresponding to them, the next step is a fixation analysis. The fixation anaylsis dismisses smooth pursuit and unstable data. This is sensible because eye traces corresponding to reading are nearly exclusively composed of fixations and saccades. Hence, periods of the gaze point sequences that do not fall properly into fixations and saccades are very unlikely to reflect reading and can be ignored.

The relevant data are now available as several sequences consisting of fixations, saccades that connect the fixations and—eventually—blinks. One such sequence can be written as

$$Q = (F_1, F_2, B_1, F_3, ..., B_m, ..., F_n)$$

where $F_1, ..., F_n$ stand for fixations each represented by a 4-tuple $F_i = F(x_i, y_i, d_i, t0_i)$ consisting of its mean coordinates (x_i, y_i), its duration (d_i) and its start point in time $(t0_i)$. $B_1, ..., B_m$ stand for blinks which have no features. Each two consecutive fixations F_i, F_{i+1} with no blink in between them are connected by a saccade which starts at x_i, y_i at the time $t0_i + d_i$ and ends at x_{i+1}, y_{i+1} at the time $t0_{i+1}$. We now translate this sequence to a form more suited for further processing:

$$\begin{aligned} Q' = \quad & (F(d_1), S(x_2 - x_1, y_2 - y_1), F(d_2), \\ & S(x_3 - x_2, y_3 - y_2), ..., B, ..., F(d_n)) \end{aligned}$$

where we only represent the duration of each fixation (d_i), the size of each saccade $(x_2 - x_1, y_2 - y_1)$ and the blinks. We simply disregard the fixations start points in time. We call this form of fixation saccade sequence the relative form. The presented models will make use of this relative sequence representation.

3 The Hidden Markov Model

A Hidden Markov Model (HMM) is similar to a finite state automaton in which the transitions as well as the outputs are probabilistic. An HMM is characterized by its transition probability distribution and its emission probability distributions. The transition probability distribution gives the probability for a transition between each two states of the model. The output produced by a state (its emission) is subject to a stochastic process characterized by the emission probability distribution of that state. This distribution assigns a probability to each possible emission. HMMs are nowadays a widely used tool in many signal processing domains [4].

In our case, the observations are a set of fixation-saccade sequences in the relative form (as introduced in section 2.1). We assume that these observations can be well described as being produced by a Markov process with hidden states. We assume furthermore

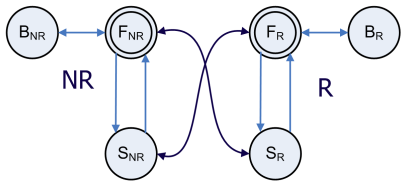

Figure 1: The 6-State Hidden Markov Model for partitioning fixation-saccade sequences into reading and non-reading.

that there is at least one distinguished state in that process; called *reading*. We know that there are other processes involved but we can not say something about their characteristics or about their number. We therefore group them in a state we call *non-reading* simply because this is a straight forward way to proceed. We assume that all possible transitions between these two states can occur. These two states, *reading* and *non-reading*, have to account for both, fixations and saccades We therefore model them internally as cycles consisting of a fixation state and a saccade state. Additionally we introduce two states to model blinks. All in all, we assume the model depicted in figure 1.

Its interpretation is as follows. F_R and F_{NR} model the fixations in the reading and non-reading case whereas S_R, and S_{NR} model saccades. Accordingly, the probability distributions (PDs) of F_R and F_{NR} are PDs over all possible fixation durations and the PDs of S_R and S_{NR} go over all possible saccade vectors (x,y). The two states B_R, B_{NR} account for blinks. However, for our application, these states turned out to be of rather little importance—if we take them (and the handling of blinks in the preprocessing step) away the classification performance of the model changes only very little.

The cyclic form of the two symmetric parts of the Hidden Markov Model simply reflects the fact that a saccade is always followed by a fixation and that every fixation, except the last, is followed by a saccade. The transitions from S_{NR} to F_R (we write that as: (S_{NR}, F_R)) and from S_R to F_{NR} model the changes from reading to non-reading and visa versa. These transitions will generally have a low probability compared with the probabilities for (S_R, F_R) and (S_{NR}, F_{NR}) because a reading fixation is more likely to be followed by another reading fixation than by a non-reading fixation and visa versa.

To prevent the model from over fitting the data by modeling correlations between Δx and Δy components of saccades, which we assume to be independent of each other, we constrain the emission PDs of the saccade states S_R and S_{NR} to be of the form $P_S((\Delta x, \Delta y)) = P_{S_x}(\Delta x)P_{S_y}(\Delta y)$, where P_{S_x} P_{S_y} are separate probability distributions for the Δx and Δy components of saccades.

All of the models emission PDs are discrete probability distributions [2] because the fixation durations as well as the saccade lengths are only available from the recorded data as discrete quantities; the former as multiples of 20 milliseconds the latter as screen pixels.

Given the depicted model, a set of model parameters (transition probabilities and emission PDs) and a certain fixation-saccade sequence, the most probable state path that could have produced the sequence can be computed using the Viterbi algorithm [4]. This path partitions the sequence into parts modeled by R-states and parts modeled by NR-states. It therefore classifies each fixation as either reading or non-reading. The remaining problem is to obtain a suited set of model parameters. This can either be done in a supervised fashion, by learning from already partitioned training data, or unsupervised, without using already partitioned training data. These approaches are described in the next two sections.

[2] It seems as if continuous probability densities would be more appropriate because of the continuous nature of the durations and saccade lengths on the one hand and because of mathematical elegance on the other hand. However, it has to be considered that durations and saccade lengths are in practice not continuous but discrete and rather coarse grained quantities and therefore are well accounted by discrete densities. Thus, we decided to use discrete emission densities.

3.1 Obtaining Model Parameters from Labeled Data

If we have preprocessed eye tracking data of the relative form $Q' = (F(d_1), S(\Delta x_1, \Delta y_1), F(d_2), S(\Delta x_2, \Delta y_2), ..., F(d_n))$ and a state sequence $L = (s_1, s_2, ..., s_{3n})$, where $s_i \in \{S_R, F_R, S_{NR}, F_{NR}\}$, which says that the first fixation $F(d_1)$ has been produced by state s_1, the first saccade $S(\Delta x_1, \Delta y_1)$ by s_2 and so forth, estimating the optimal model parameters is easy.

Let us assume that $F(d)$ is produced q times by the state F_R and that the state F_R occurs t times in L. Given that we have no more information, the probability that $F(d)$ is produced by F_R is $P(F(d)|F_R) = q/t$. If we denote the number of times a certain state s occurs in L with t_s and the number of times this state produces the emission e with $g_s(e)$ the probability that e is emitted by s is given by $P(e|s) = g_s(e)/t_s$. The transition probabilities are estimated similarly. If the number of times s' follows s in L is given by $n_{ss'}$, the probability of a transition from s to s' is given by $P(s_{i+1} = s'|s_i = s) = n_{ss'} / \sum_{s'} n_{ss'}$.

If such an estimation is done on a finite training set the resulting emission probability densities are—because of the stochastic nature of the underlying process—only approximations to the real densities. They will be superpositions of the real densities and a noise component, the noise in general being bigger as the amount of training data decreases. Thus, given only little data, this leads to an overfitting of the training data at the expense of a worse generalization. In the worst case, it can even happen that certain emissions do not occur in the training data at all. The probability for those emissions will be set to zero. That would lead to an inability of the model to account for sequences containing them. Scaling the whole density by setting $p(e) = p(e) + \gamma$ and then renormalizing it, guarantees each emission probability to be greater than zero[3] (it guarantees furthermore that an emission occurring in the data always has a higher probability than one which does not occur).

The problem of noise overlaying the densities can be tackled further: for example, by fitting the estimated probability densities to a function of a simpler type (e.g. a polynomial) to eliminate the noise. We tested this method on a smaller data set as well as on the whole data set. We did not fit the PDs with a polynomial but with a nonlinear combination of sigmoidals realized by a 3-layer perceptron network (often called back propagation network) with sigmoidal activation functions because this gave better overall results [4]. Used on the smaller training

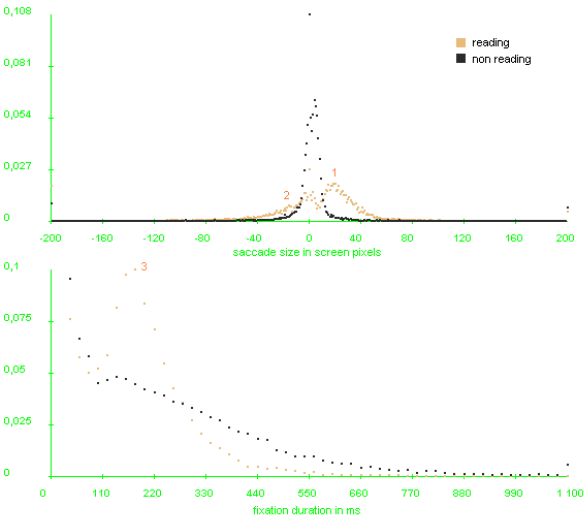

Figure 2: The PDs of the states of the 6-State Hidden Markov Model as estimated from the training set. The upper graph shows the PDs of the Δx-components of *reading* (orange) and *non-reading* (gray) saccades. Δx (in screen pixels) is plotted against the respective probability. The peak at (1) corresponds to the most common type of (forward) reading saccade. The peak at (2) corresponds to a quite frequent type of backward going saccade. The lower graph shows the PDs of F_R (orange) and of F_{NR} (gray). The duration (in milliseconds) is plotted against the respective probability. The peak at (3) corresponds to the typical reading fixation duration.

[3] We used a γ value of 0.0001.

[4] The neural network used for fitting had 7 hidden neurons and was trained by minimizing a standard quadratic error func-

set, this method gives rise to a significant increase in generalization performance whereas it does not yield a significant increase in generalization performance when using the whole training set. To evaluate the classification capacities of the model trained in supervised fashion the set of labelled data was divided into two disjoint sets, a training and a validation set. The model parameters were estimated from the training set, then the model was tested on the validation and the training set. We achieved a precision/recall performance of 0.88/0.87 on the validation set. These numbers are discussed in section 6. The PDs over fixation durations and saccade's Δx-components lengths of the reading and non-reading states as estimated from the training set are shown in figure 2.

Finally we investigated how much each of the features fixation duration, saccade Δx-component and saccade Δy-component contributes to the classification performance. The result is that the x-components of the saccades contribute much more than the fixation durations which in turn contribute more than the y-components of the saccades. Using only the x-components yields a precision/recall performance of around 0.8/0.8, including fixation durations gives 0.86/0.84, adding also the y-components finally yields 0.88/0.87.

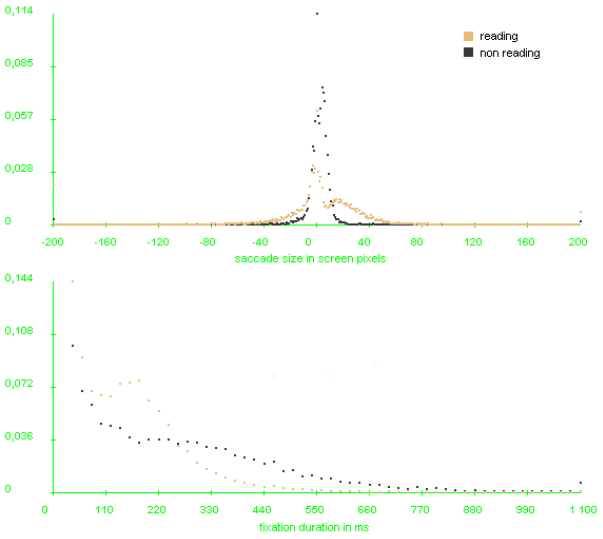

Figure 3: The PDs of the states of the 6-State Hidden Markov Model as estimated in the unsupervised case. The upper graph shows the PDs of the Δx-components of *reading* (orange) and *non-reading* (gray) saccades. Δx (in screen pixels) is plotted against the respective probability. The lower graph shows the PDs of F_R (orange) and of F_{NR} (gray). The duration (in milliseconds) is plotted against the respective probability.

3.2 Obtaining Model Parameters from Unlabelled Data

An interesting aspect of Hidden Markov Models is that efficient algorithms are available to optimize an HMM's parameter set with respect to certain observations even if the underlying state sequence is not known. If a certain HMM topology is assumed and a set of observation sequences is known these algorithms allow us to find a "good" set of PDs to account for the observation sequences. They can be used to form hypotheses about the data if one can assume that the process which produced the data can be described well as a Markov process with a certain topology. The usage of such an approach will now be demonstrated. The fact that we have information about how the PDs of the states should look like and about how the fixation saccade sequences will fall into reading and non-reading puts us in the position to evaluate it.

Given a certain HMM topology and a set of saccade-fixation sequences $O_1, O_2, ..., O_n$ we can find a set of model parameters that locally maximize the likelihood of $O_1, ..., O_n$ with respect to the model. This is typically done in an iterative manner. We start with an arbitrarily chosen set of model parameters and repeat an expectation and a *reestimation step* until the total change in likelihood for the sequences $O_1, O_2, ..., O_n$ is very small or a maximum number of iterations is exceeded. In the *expectation step* the expected number each transition and each emission of the HMM is used when modeling $O_1, ..., O_n$ is calculated. During the *reestimation step*

tion using the backpropagation algorithm [5].

the the model parameters are reestimated on the basis of the expected values. This is done in a way that guarantees the proportion of each two probabilities of the new parameter set to be equal to the proportion of the respective expectation values.

This algorithm is often called forward-backward or Baum-Welch algorithm. A detailed description of it can be found in [4]. After each iteration, the likelihood of $O_1, ..., O_n$ either stays the same (in this case the model parameters define a local maximum (or saddle point) of the likelihood function) or increases. It can be shown that the model parameters converge to a local maximum (or saddle point) of the observation sequence likelihood function.

Using the forward backward algorithm, we can find a set of PDs that locally maximizes the probability of our fixation saccade sequences. The hope is that, in the end, these estimated PDs will resemble the PDs of the Markov process that we assumed to be a good description of the underlying process. If that happens the PDs of the "trained" model will reflect *reading* and *non-reading*[5] and the resulting model will be able to partition the observation sequences into *reading* and *non-reading*.

We tested this approach with the training set used in section 3.1. The resulting model achieved a precision/recall performance of 78/86 (compared to 0.88/0.87 in the supervised case) when tested on the training set. The resulting PDs of the *reading* and *non-reading* states are shown in figure 3. They are quite similar to the PDs estimated in section 3.1, although not as pronounced.

4 On line Applications

An important feature of our classification approach is that it can be used on line while recording the eye tracking data. This allows for automatic reaction whenever the subject is reading and makes new experimental designs possible. One example would be to change the stimulus every time the subject is reading. We will now depict how and how good such an on line classification can be done using the HMM depicted in section 3.

Given that the gaze points of a subject are known up to time t a fixation analysis can be performed and the sequence in relative form can be obtained exactly as in the off line case. It may happen that the last fixation is still ongoing at time t. In that case, we wait with the classification until its duration stays constant. If the last fixation is complete, we have a partial fixation saccade sequence up to time t. This sequence can now be partioned into reading and non-reading in the same way as we have done it in the off line case. It is to be expected that the classification performance will decrease due to the lack of information about future fixations. Furthermore, one would expect a slight latency, at least in switching from reading to non-reading. We simulated the on line application of our approach for a certain sequence, by considering a series of partial sequences. Each partial sequence is a section of the original sequence from the sequence start up to a certain time t where t runs from the sequence start to the sequence end. After a fixation analysis, we let the HMM classify the partial sequence and stored the label it assigned to the last fixation of the sequence. In that way all fixations were classified and we obtained a partition of the whole sequence. An example of such a partition in contrast to a partition obtained in the off line case is shown in figure 4. A slight latency of the on line classification can be seen. It amounts to 2-3 fixations in average. We have tested for precision/recall performances exactly like we did in the off line case. On the validation set we found a precision/recall of 0.8/0.8 (compared to 0.88/0.87 in the off line case). For some applications, the precision/recall performance may not be sufficient. In these cases, the quality of the experimental data can be ensured using the off line classification after the experiment to eliminate data that has been incorrectly classified on line. It may be possible to reduce also the latency, which could, for example, be a problem if very short reading phases should be detected, by providing additional information (e.g. if the subject is looking at text) and thus making classification easier.

[5] If we start the estimation with arbitrary model parameters we will not know in advance which state finally will reflect *reading* and which *non-reading*—we have to decide this later on by looking at the states. As we know that in our data *reading* occurs less often than *non-reading* we can do this automatically by chosing the state that occurs least to be *reading*.

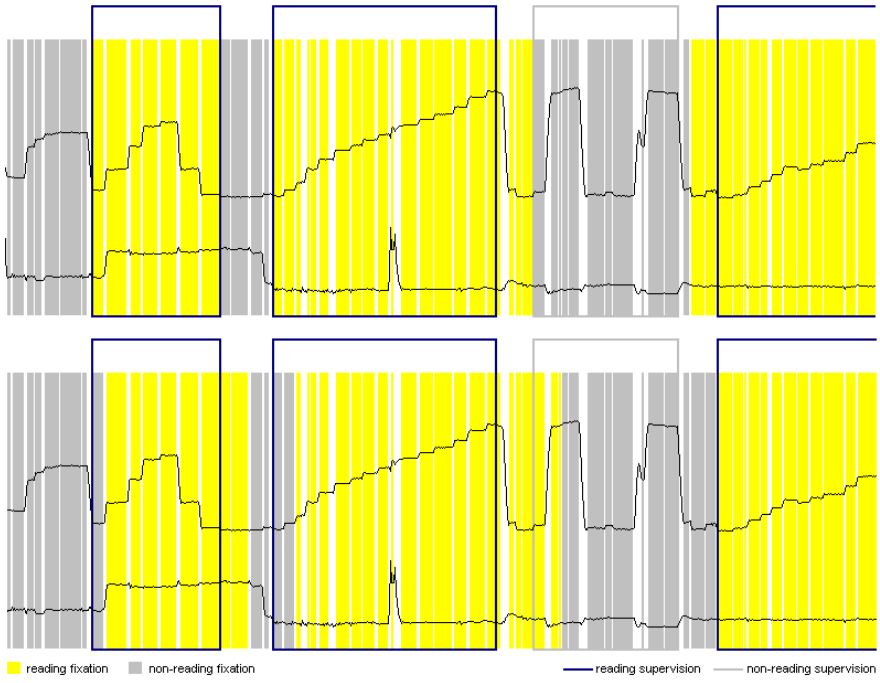

Figure 4: The classification of a short gaze sequence by the 6-State Markov Model in off line (top) and on line (bottom) mode. Note the latency in the on line mode. It amounts to 2-3 fixations in average.

5 A Neural Network Approach

A drawback of the HMM Model seems to be that it only models probability distributions conditioned on whether or not reading is taking place. The probabilities for the fixation durations or saccade lengths depend only on the actual state of the model. It accounts only indirectly for neighboring parts of the fixation saccade sequence. It seems sensible to ask for a way to account for these neighboring parts in a rather direct way [6]. The PDs of the *reading* and *non-reading* states should be conditional. The probability for a certain fixation to belong to *reading* or *non-reading* respectively should depend directly on the neighboring fixations. This is rather complicated to achieve using a Hidden Markov Model but can easily be done with a neural network. We tested such an approach using a 3-layer neural network where we take the actual fixation as well as the neighboring fixations in a window of fixed size (we used 4 fixations before and 4 after the actual fixation) as input. The network was trained[7] on the same training set as the HMM. Its performance in the precision/recall tests was about the same as the performance of the 6-State HMM. However, the training took much longer (several hours compared to seconds needed for "training" the HMM). Additionally, the resulting network is hardly as transparent as a Hidden Markov Model where we can directly access

[6] Consider for example the case of return sweeps: Two subsequent return sweeps are highly unlikely. The 6-State HMM can not account for that, since the only information available for classification is whether the previous observation belonged to reading or not—whether it was a return sweep or not is hidden. The neural network on the other hand can account for that.

[7] The network had 15 hidden neurons. We used gradient descent and backpropagation to minimize a standard quadratic error function.

Classification Method	Performance on Training Set	Performance on Validation Set
6-State HMM	0.92/0.91 (0.93)	0.88/0.87 (0.90)
6-State HMM, simulated on line condition	0.83/0.82 (0.88)	0.8/0.8 (0.86)
6-State HMM, unsupervised training	0.78/0.86 (0.87)	—
Neural Network	0.89/0.88	0.86/0.86

Table 1: The Classification Performances measured against the classifications of a human expert. The slash-separated numbers are precision/recall values whereas the number in brackets says how many percent of all fixations in the data set were classified correctly.

the PDs and where we can easily understand how the classification is done. Furthermore, a network of this type allows only for supervised training (although good unsupervised training may be possible using self organizing neural networks).

6 Results

Table 1 shows the performance as precision and recall[8] with respect to reading. The total percentage of correctly classified fixations is also shown. Particularly important are the values in the third column (Performance on Validation Set), because they say how good the model generalizes—how it performs on data that it has not seen before—and thus estimate the performance on data that has not been labelled at all. According to that column every 12th fixation is classified incorrect in the supervised case. These errors mostly occur at the borders of reading or non-reading phases respectively. Very often the models labeling starts or ends a few fixations earlier or later than the supervisor labeling. Total misses of reading phases and the labeling of reading during a non-reading phase are very rare. All in all we had the impression that the models labellings were very accurate and that a human expert could not do much better.

[8]Precision: How many of the fixations the model classified as reading where actually labelled as reading fixations by the supervisors; Recall: How many of the labelled reading fixations did the model classify as reading

An evaluation, as it is done here, is a bit problematic because it is not absolutely clear what pattern of fixations and saccades can be called reading and what can not. Furthermore, it is difficult to give definite borders of reading and non-reading phases (That could, at least partly, serve as an explanation of the models labeling errors at borders of reading and non-reading phases). Thus, even though only very clear cases were labeled in the evaluation and validation sets, a certain degree of subjectivity remains.

7 Concluding Remarks

Reading research with eye movement measurements has existed since the 1920s. Today, there exist several generative models that predict eye movements during reading, for instance the EZ-reader [6] or the Swift model [7]. These models generate fixations and saccades that would typically occur when a certain text is read. The Hidden Markov Model approach presented here also models reading, but from a stance of descriptive statistics and without consideration of the stimulus. It is capable of predicting whether a given fixation and saccade sequence reflects reading or not. That makes it a useful tool in the analysis of experimental data. It is an alternative to ROI analysis, which may fail in cases where looking at a text not necessarily means reading it. Our approach is also fast and can be used on-line, during recording, with considerable performance. This opens up possibilities for new experimental paradigms.

References

[1] M. Hannus J. Hyönä. Utilization of illustrations during learning of science textbook passages among low- and high-ability children. *Contemporary Educational Psychology*, 24:95–123, 1999.

[2] K. Holmqvist, V. Johansson, S. Strömqvist, and Å. Wengelin. Studying reading and writing online. In S. Strömqvist, editor, *The diversity of languages and language learning, Lund University, Centre for languages and literature*. 2002.

[3] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124:372–422, 1998.

[4] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] E. D. Reichle, K. Rayner, and A. Pollatsek. The e-z reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and Brain Sciences*, 26:445–526, 2003.

[7] R. Engbert, A. Longtin, and R. Kliegl. A dynamic model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research*, 42(5):621–636, 2002.