Proceedings of the NIPS 2005 Workshop on Machine Learning for Implicit Feedback and User Modeling

Workshop at NIPS 2005, in Whistler, BC, Canada,
on December 10, 2005.

http://www.cis.hut.fi/inips2005/

Kai Puolamäki and Samuel Kaski, editors

Otaniemi, May 2006

ISBN 951-22-8219-4 (printed version)
ISBN 951-22-8220-8 (electronic version)

Otamedia
Espoo 2006

Workshop on Machine Learning for Implicit Feedback and User Modeling

In Whistler, Canada, on December 10, 2005.

Program Committee

Samy Bengio, IDIAP
Helene Hembrooke, Cornell
Thorsten Joachims, Cornell
Samuel Kaski, Helsinki University of Technology
Ilpo Kojo, Helsinki School of Economics
Petri Myllymäki, University of Helsinki
Kai Puolamäki, Helsinki University of Technology
Kari-Jouko Räihä, University of Tampere
John Shawe-Taylor, University of Southampton

Organizers

Samuel Kaski and Kai Puolamäki
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400
FI-02015 TKK, Finland

The workshop was a core event of the PASCAL EU Network of Excellence.
The workshop was part of the NIPS 2005 Workshop Programme.

Table of Contents

Preface

The workshop, organized in the NIPS Workshop programme on December 10, 2005, was arranged to gather together machine learning researchers interested in a new promising and challenging research area. Some of the papers introduced best-performing solutions of a PASCAL Eye Movement Challenge, of inferring intent of users based on eye movement signals.[1] The rest broadened the scope towards other implicit feedback signals, and towards more general problems of user modeling.

The tasks require advanced signal processing and feature extraction, and dynamic machine learning models. Several branches of machine learning are applicable, and we expect eye movement data to become a new challenging testbench for machine learning algorithms.

The Eye Movement Challenge was organized for the PASCAL network but participation was open to all. We made eye movement data available, and the objective was to predict from eye movement data whether a reader finds a text relevant. The scientific goals of the Challenge were to advance machine learning methodology, to find the best eye movement features, and to learn of the psychology underlying eye movements in search tasks. The setting and the data have been described in more detail in the technical report[2], reproduced for convenience in this proceedings volume.

The challenge consisted of two different parts. In the first, the eye movements had been preprocessed into a sequence of feature vectors, and the task of the participants was to predict, for each sequence, whether the read text was relevant or not. In machine learning terms, the task was binary classification of multidimensional sequences.

Several kinds of machine learning methods were tried on the problem. An ensemble of multilayer perceptrons of Pfeiffer et al. was the winner. They re-computed a new set of features, which may account for part of the success. Decision trees performed well too; no paper on them was presented in the workshop, however. Conditional random fields of Do and Artières were among the best, followed by a very different methodology, symbolic approaches based on finite state machines of Largeron and Thollard. Several additional, less well performing methods submitted to the competition remain undocumented, since papers were not submitted to the workshop.

It is difficult to draw conclusions on the relative performance of the various

[1] The Challenge has a web site at http://www.cis.hut.fi/eyechallenge2005/.

[2] Jarkko Salojärvi, Kai Puolamäki, Jaana Simola, Lauri Kovanen, Ilpo Kojo, Samuel Kaski. *Inferring Relevance from Eye Movements: Feature Extraction.* Helsinki University of Technology, Publications in Computer and Information Science, Report A82. 3 March 2005.

methods since many of the applications were only quick feasibility studies. The difference in the performance of the best performing documented methods was only about two percentage units, of which the new feature extraction performed by the winners may account for a major share. The winning entry obtained 72.3 % classification accuracy, while the dummy model, used as a baseline, which assigned everything to the largest class would have given 46.5 % accuracy in the test data set. The choice of the machine learning methods is probably not as crucial as the choice of appropriate features, a lesson often learned in machine learning studies.

In the second part or subchallenge, the participants were given the raw eye movement trajectory, but the task was still the same: Predict, based on the trajectory, whether the read text was relevant. This competition was clearly more laborious, and attracted only three participants. Of these, the clear winner was Lepola, with an innovative methodology that mimics neural microcircuits.

In addition to reporting on competition results, the workshop was intended to broaden the discussion to other applications of eye tracking, and other related research on user modeling. This was achieved through the invited talks, and the contributed papers by Dorr et al., where they predict eye movements in order to guide them with suitable visual stimuli.

Finally, we wish to thank the programme committee for the refereeing work, Dr. Greg Edwards for the enthusiastic PASCAL invited talk, Dr. Cristian Sminchisescu and Dr. Wei Zhang for the two other invited talks, all other speakers and participants of the challenge, Mr. Jarkko Salojärvi and Mr. Lauri Kovanen for hard work in organizing the challenge, and PASCAL for funding.

Samuel Kaski and Kai Puolamäki
Laboratory of Computer and Information Science
Helsinki University of Technology

Workshop Programme

In Whistler, Canada, on December 10, 2005.

MORNING SESSION 7:30–10:30

7:30	Welcome and overview	Samuel Kaski & Kai Puolamäki
7:45	Why solving eyetracking analysis issues is valuable and fun* *(PASCAL Invited Talk)*	Greg Edwards
8:45	Discriminative Models and Learning for 3d Human Motion Analysis* *(Invited Talk)*	Cristian Sminchisescu
9:00	*Coffee break*	

Presentations of best performing systems from the PASCAL Eye Movement Challenge:

9:20	Symbolic answers to an eye-tracking problem**	Franck Thollard
9:40	Conditional Random Field for tracking user behaviour based on his eye's movements**	Trinh Minh Tri Do
10:00	Predicting Text Relevance from Sequential Reading Behavior** *(Winner of Competition 1)*	Michael Pfeiffer
10:20	Wrap-up and discussion	
10:30	*End of morning session*	

AFTERNOON SESSION 15:30–18:30

15:30	Inferring Relevance from Eye Movements Using Generic Neural Microcircuits** *(Winner of Competition 2)*	Tuomas Lepola

Other talks about implicit feedback and user modeling:

15:50	Learning the users interests using the search history	Nesrine Zemirli
16:20	Predicting, analysing, and guiding eye movements**	Martin Böhme
16:50	*Coffee break*	
17:10	A user model of eye movements during visual search* *(Invited Talk)*	Wei Zhang
17:40	Wrap-up and discussion	
18:30	*End of workshop*	

* Abstract is included in this proceedings volume.
** Extended abstract is included in this proceedings volume.

Why solving eyetracking analysis issues is valuable and fun*

Greg Edwards
Eyetools Inc.

Abstract

Eyetools was born in 2000 out of the Stanford University Advanced Eye Interpretation Project. After seeing the business value of eyetracking resulting from the Stanford-Poynter Project, a collaborative study between the Poynter Institute and Stanford University's Department of Communications around the viewing of online news sites, founder Greg Edwards spun out Eyetools. Since then, Eyetools' pioneering work in inferring mental state from eye movements and visualizing eyetracking data has led to several key patents in the area, and has enabled eyetracking to be put into use more easily by an ever expanding number of companies and people. Eyetools' roots in Human-Computer Interaction began in 1995

*PASCAL Invited Talk

Discriminative Models and Learning for 3d Human Motion Analysis*

Cristian Sminchisescu
TTI-C
University of Toronto
Rutgers University

Abstract

I will discuss discriminative learning algorithms for estimating 3D human motion in monocular video sequences. The complexity of the problem stems from the high-dimensionality of the human (joint angle) state space and from the unknown and variable nature of human surface and appearance parameters in many real scenes. Depth ambiguities, image clutter and occlusion further complicate matters. While this problem has been traditionally approached using the powerful machinery of generative models, the main emphasis of this talk will be on an emerging class of complementary discriminative temporal estimation models. These can be viewed as up-side down, mirrored versions of the classical temporal chains used with Kalman filtering or Condensation. But rather than inverting a generative imaging model at runtime, we will learn to cooperatively predict complex local image-to-state mappings, using Conditional Bayesian Mixtures of Experts. These are embedded in a probabilistic temporal framework based on Discriminative Density Propagation in order to enforce dynamic constraints and allow a principled propagation of uncertainty. The models are trained using a human motion capture database and a 3D computer graphics human model to synthesize pairs of typical human configurations together with their realistically rendered 2D image silhouettes. To demonstrate the algorithms, I will present empirical results on real and motion capture-based test sequences.

This is joint work with Atul Kanujia, Zhiguo Li and Dimitris Metaxas.

*Invited Talk

Symbolic answers to an eye-tracking problem

Christine Largeron
EURISE*
largeron@univ-st-etienne.fr

Franck Thollard
EURISE*
thollard@univ-st-etienne.fr

Abstract

We provide in this article experiments made on the eye-tracking challenge proposed by the PASCAL European network. We concentrate here on symbolic approaches mainly based on finite states machines. Our experimental study opens many questions mentioned as a conclusion.

1 Introduction

We address in this paper some experiments made on a shared task proposed by the PASCAL[1] network and which concerns proactive information retrieval [4]. In this task a reader is given a question and 10 sentences, one of them being the correct answer to the question, 4 being relevant and 5 irrelevant. Some information such as the scheduling of the reading or the pupil diameter of the eye of user are stored. During the learning process the machine is given the reading features and the label of the sentences (2 for correct answer, 1 for relevant, and 0 for irrelevant). At evaluation time, the machine is asked to label the sentences. More information on the task together with the data sets can be found at the challenge web page: `http://www.cis.hut.fi/eyechallenge2005/`.

We analyzed the data using different approaches. We first built a graphical interface of the data from which we get a visual rendering of the user behavior. We then used some statistical approaches in order to find relevant features. We then applied decision trees (C5) to handle numerical and categorical features. In order to take into account the behavior of the user, we finally transformed the data in a symbolic form and used syntactic models.

2 Analysis of the data

2.1 Graphical Data Interface: GDI

We built a graphical interface of the data (GDI) – see figure 1 – in order to see what words the user are reading and in which order. On the GDI, we can select a question (or assignment) and a number that allows to tune the time unit. The words of each of the 10 answers are drawn in a color that corresponds to their labels. As the simulation starts, the word being read is colored in a different color, showing the scheduling of the reading.

*EURISE, Jean-Monet University, 42023 Saint-Etienne Cedex 2 France

[1]PASCAL stands for Pattern Analysis, Statistical Modelling and Computational Learning.

Figure 1: Eye tracking Graphical Data Interface

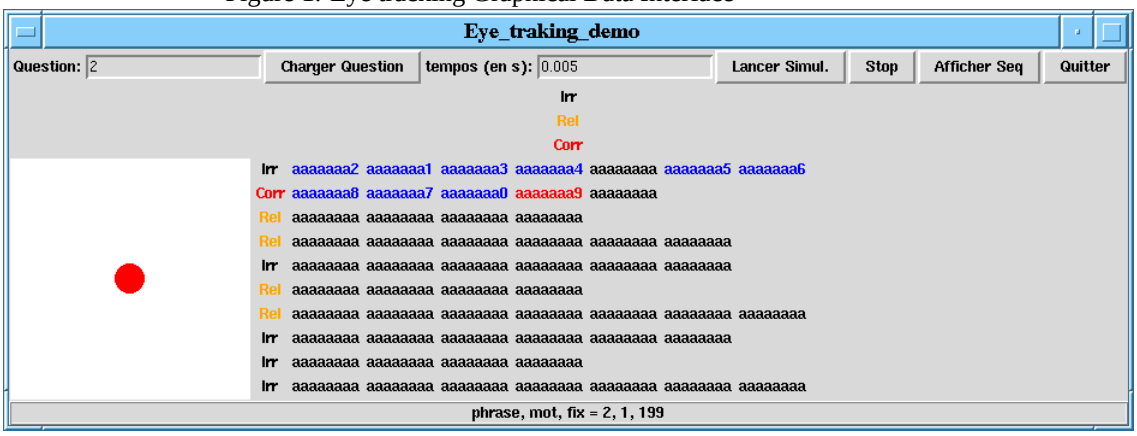


Table 1: Correlation rates

Correlation rates above 0.7	Overall	Label 0	Label 1	Label 2
PrevFixPos – FirstSaccLen	0.794	0.798	0.796	0.785
PupilDiamLag – PupilDiamMax	0.772	0.709	0.721	0.86
MeanFixDur – totalFixDur	0.696	0.711	0.696	0.681
fixcount – firstPassCnt	0.674	0.813	0.778	0.496

On the left hand side of the sentences, a circle is drawn in the color of the label of the word being read; Its size changes according to the pupil diameter.

This GDI allows us to see that the users, almost always, finish the parsing of the 10 answers on the correct one. Labeling the last read sentence as label 2 performs a precision and recall around 92.5% on the validation set.

2.2 Statistical analysis of the data

When facing a new problem, a first natural step could consist in understanding the data. We therefore made a computation of correlation rates and principal component analysis.

2.2.1 The correlation rate

We first computed the correlation rates between the numerical variables. As shown in table 1, the rates were not very high: only very few correlation rates are above .7. The highest value is obtained for prevFixPos and firstSaccLen. But, when we considered only the records corresponding to each label, the rates did not stay constant. For instance the correlation between fixcount and firstPassCnt is only equal to 0.496 for the label 2 when is equal to 0.674 on the training set. It seems then not possible to reduce the number of features by leaving out highly correlated features.

2.2.2 The principal component analysis

We continued our study with Principal Component Analysis (PCA) in order to find out whether there were clusters in the data. We used centered data to preserve the distance between the records instead of normalized data as usually.

Table 2: PCA and C5

PCA	C5	Comp. nb.	Eigen val.	% of var.	cumul. var.
fixcount	fixcount	1	3807987.65	85.249	85.249
firstpasscnt	firstpasscnt	2	465286.091	10.416	95.665
prevfixdur	P1stFixation	3	74205.070	1.661	97.326
firstfixdur	P2stFixation	4	44286.255	.991	98.318
firstpassfixdur	prevfixdur	5	26521.484	.594	98.911
nextfixdur	firstfixdur	6	14507.983	.325	99.236
v11n	firstpassfixdur	7	7774.952	.174	99.410
lastsacclen	nextfixdur	8	7009.023	.157	99.567
prevfixpos	firstSaccLen	9	5245.223	.117	99.685
landingpos	lastsacclen	10	5107.055	.114	99.799
leavingpos	prevfixpos	11	4116.011	9.214E-02	99.891
totalfixdur	landingpos	12	3881.247	8.689E-02	99.978
meanfixdur	leavingpos	13	984.843	2.205E-02	100.00
nregressfrom	totalfixdur	14	.297	6.659E-06	100.00
regresslen	meanfixdur	15	.281	6.283E-06	100.00
regressdur	nregressfrom	16	4.971E-02	1.113E-06	100.00
puplidiammax	regresslen	17	3.716E-02	8.320E-07	100.00
pupildiamlag	nextWordRegress	18	1.602E-02	3.586E-07	100.00
timeprtctg	regressdur	19	4.204E-04	9.411E-09	100.00
	puplidiammax				
	pupildiamlag				
	timeprtctg				

Table 3: Confusion Matrix for the C5 algorithm

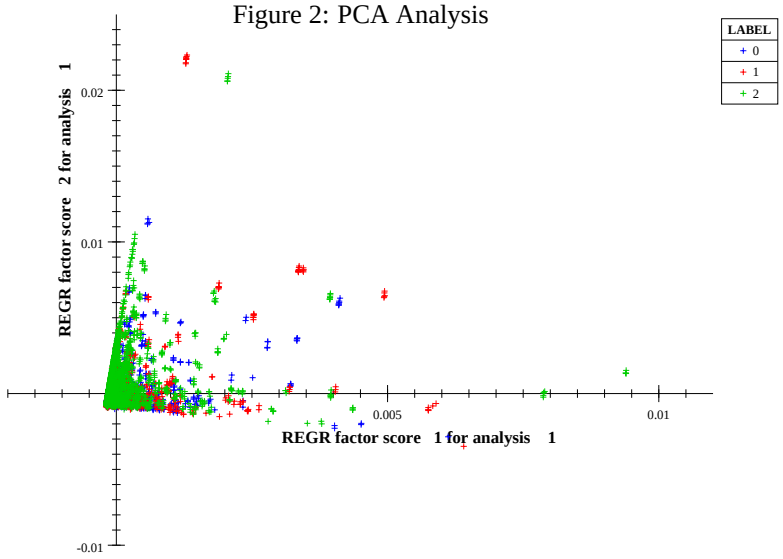
True / Predicted	0	1	2
0	377	196	9
1	255	226	2
2	5	6	138

Table 2 (left, first column) gives the list of the variables used in the PCA. As we can see in table 2, (right) factor 1 accounts for 85.25% of the variance, factor 2 for 10,42% , and so on. The last column contains the cumulative variance extracted. According to the Cattell's criterion [1], we could retain two factors to summarize the data set. Judging from the projection of the training set on the two principal axes, it was not possible to separate the three clusters. The identification of each element by its label on the PCA plot (Figure 2) confirmed this result.

These conclusions lead us to consider a large set of variables, numerical and categorical, and to use C5 classifier designed by Quinlan[2] to handle the features given in Table 2 column 2. Besides that, all the records corresponding to the last sentence read have been excluded. Following the rule deduced from the GDI, the decision for these records is correct answer (label = 2). Results over the evaluation set are reported in Table 3. As expected by the preliminary analysis, results are not extremely high insofar as accuracy is 61.04% on the evaluation and 60.57% on the test set. We thus decided to model the user's behavior.

[2]See `http://www.rulequest.com/see5-info.html` for more details on the C5 algorithm.

Figure 2: PCA Analysis



3 Probabilistic finite state models

The idea of the approach consists in modeling the reading of the user as a path in a finite states machine. We applied the following strategy:

1. discretizing the data,
2. splitting the training set in order to have a learning set for each target label,
3. building three models, one for each label,
4. guessing the label according to each model, and the fact that there is exactly 4 (respectively 5) relevant (respectively irrelevant) sentences in each assignment.

3.1 Discretizing the data

The aim of the discretization is to model the behavior of the user as a string.

We decided to describe an eye movement and its intensity by a pair of characters. We built, by hand, a 9 words vocabulary: B0 B1 B2 E F0 F1 F2 Q0 Q1. Except for the letter E which models the end of the reading, each symbol is composed by two components, a letter indicating an eye movement and a number modeling how important the movement was. B stands for Backward reading, F for Forward reading, and Q for Quitting the sentence.

3.2 Building the models

From this coding we built three multisets of strings (one for each label) of the form:

F0 F0 F0 Q0
F0 F0 F0 Q1
F0 F0 F0 Q1
F0 Q1
F0 F0 Q1
F0 Q1 F0 F1 F1 B0 Q0 B0 E
F0 Q1

Table 4: Perplexity of the models (the lower, the better).
0 – 1 means "learning on train 0 and testing on validation 1"

parameter	0–0	0–1	1–0	1–1
0.05	5.13839	6.6329	5.32001	6.4469
0.01	3.46395	4.69896	3.58998	4.55078
0.002	2.94324	4.68004	4.28921	8.33305
0.005	2.99874	4.64709	3.10137	4.3912
0.0005	8.67323	22.7453	8.94031	21.9032

Table 5: Confusion Matrix for the different approaches

	Automaton model			3-gram model			3-gram - ad-hoc		
True / Predicted	0	1	2	0	1	2	0	1	2
0	438	135	9	421	152	9	381	194	7
1	267	214	2	248	233	2	183	296	4
2	2	9	138	2	9	138	2	9	138
Overall Accuracy	65%			65%			67%		

Since the sets are multisets, we decided to take into account this information by building probabilistic models. We used two kind of models: smoothed trigram and probabilistic automata. Each of these models provides a probability distribution over $\Sigma^\star$, Σ being the vocabulary.

The algorithm for inferring probabilistic automata [3] has a tuning parameter. We usually get the value of the parameter by minimizing the perplexity [2] (or equivalently maximizing the average of the probabilities the inferred automaton provides) on a held out set. Table 4 provides the perplexity obtained by the models on different data. Column 0–1 means: train the model on irrelevant sentences and evaluate it on a held out set of relevant ones.

According to table 4, we decided to use the parameter 0.0005 in order to maximize the margin between label 0 and 1.

3.3 Guessing the label

For we have a quite good rule for label 2 (extracted from the GDI), we decided to first set the label 2 for the sentences on which the user finishes the reading (*i.e.* sentences that contain E in their coding) and then consider a two class problem.

On table 4 we can see that the model built on relevant sentences is not good as it predicts better irrelevant sentences than relevant ones. We thus decided to consider only the model built on irrelevant sentences (*i.e.* sentences labelled 0) and accordingly set label 0 to the 5 more probable sentences according to the model 0 and label 1 to the other ones. The performances of this strategy is given in table 5, left, and performs a global accuracy of 65 % on the validation set. We did the same experiments using a 3-grams model which obtains equivalent performances (65%, table 5, center).

Note that a specific method has been designed by hand by C. de la Higuera for guessing the label given the models. This ad-hoc strategy raises the performances to 67% (table 5 right).

4 Conclusion and further works

In this article, we proposed to use symbolic approaches in order to tackle the eye tracking problem. We identified different steps: building a symbolic coding of the data, inferring syntactic models and guessing the final labels.

We proposed different methods for each step: automatic `vs` hand-made building of the coding, building probabilistic automata `vs` 3-grams, general method for guessing and ad-hoc method. Even if the results are not as bad as compared to the other methods, we now face more questions than answers:

Automatic building of the coding: building the coding automatically is a problem in itself. We tried to build the coding automatically using the rules provided by the C5 algorithm but the preliminary results were very disappointing (*i.e.* $\sim$ 55% of accuracy on the validation set). We thus built the coding by hand keeping in mind the following rules:

- the same string must belong to only one class,
- the vocabulary must be quite small in order to avoid the ”sparse data problem”,
- the sentences of the coding must be quite short,
- the vocabulary must model/select relevant features (e.g. the E symbol that model the ”end of reading”).

In order to optimize the coding itself, it would be good to define an ”off line” quality measure of a coding, that is, in some way, quantifying the above rules.

Quality measure for the inference: as seen before, the best value for the tuning parameter for this task was not the one for which the better model – in term of prediction power– is built. We thus think that a new quality measure is needed in such a case.

Final guess of the label: in the experiments presented, we noted that the results were drastically improved when a consistent labelling is guaranteed (which means, in our case, exactly one sentence is labelled correct, 4 relevant and 5 irrelevant). Moreover, the results can be very different depending on the job done at that step. We think that some more automated work is needed here.

Following one of the anonymous reviewers who ”guess that the strengths of symbolic approach [...] might be simplicity, robustness and speed of implementation”, we would like to continue this work in that direction.

Acknowledgements The authors wish to thank Colin de la Higuera, Thierry Murgue and Jean-Christophe Janodet for fruitful discussions.

References

[1] Cattell, R. (1966) The scree test for the number of factors. *Multivariate Behavioral Research*, 1.

[2] Goodman, J. (2001) *A bit of Progress in Language Modeling*, Technical report MSR-TR-2001-72.

[3] Thollard, F. (2001) *Improving Probabilistic Grammatical Inference Core Algorithms with Post-processing Techniques*, pp 561-568, ICML’2001, Williams/Morgan Kauffman

[4] Salojrvi, J & Kojo, I. & Simola, J. & and Kaski, S.: Can relevance be inferred from eye movements in information retrieval? In (WSOM’03), Hibikino, Kitakyushu, Japan, 2003. pp. 261-266.

Conditional Random Field for tracking user behavior based on his eye's movements[1]

Trinh Minh Tri Do
LIP6, Université Paris 6
8 rue du capitaine Scott
75015, Paris, France
do@poleia.lip6.fr

Thierry Artières
LIP6, Université Paris 6
8 rue du capitaine Scott
75015, Paris, France
Thierry.artieres@lip6.fr

Abstract

Conditional Random Fields offer some advantages over traditional models for sequence labeling. These conditional models have mainly been introduced up to now in the information retrieval context for information extraction or POS-tagging tasks. This paper investigates the use of these models for signal processing and segmentation. In this context, the input we consider is a signal that is represented as a sequence of real-valued feature vectors and the training is performed using only partially labeled data. We propose a few models for dealing with such signals and provide experimental results on the data from the eye movement challenge.

1 Introduction

Hidden Markov models (HMM) have long been the most popular technique for sequence segmentation, e.g. identifying the sequence of phones that best matches a speech signal. Today HMM is still the core technique in most of speech engines or handwriting recognition systems. However, HMM suffer two major drawbacks. First, they rely on strong independence assumptions on the data being processed. Second, they are generative models that are most often learned in a non discriminant way. This comes from their generative nature, since HMM define a joint distribution $P(X,Y)$ over the pair of the input sequence (observations) X and the output sequence (labels) Y. Recently, conditional models including Maximum Entropy Markov models [1] and Condition Random Fields [2] have been proposed for sequence labeling. These models aim at estimating the conditional distribution $P(Y/X)$ and exhibit, at least in theory, strong advantages over HMMs. Being conditional models, they do not assume independence assumptions on the input data, and they are learned in a discriminant way. However, they rely on the manual and careful design of relevant features.

[1] This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

Conditional Random Fields (CRF) has been shown to overcome traditional Markovian models in a series of information retrieval tasks such as information extraction, named entity recognition… Yet, CRF have to be extended to more general signal classification tasks. Indeed, the information retrieval context is very specific, considering for instance the nature of the input and output data. Designing relevant features for such data is maybe easier than for many other data. Also, algorithms proposed for training CRF require a fully labeled training database. This labeling may be available in information retrieval tasks since there is a kind of equivalence between nodes and labels but it is generally not available in other signal processing tasks. Hence, the previous usages of CRF do not fit well with many sequence classification and segmentation tasks concerning signals such as speech, handwriting etc. Input data is rougher; it is a sequence of real-valued feature vectors without precise semantic interpretation. Defining relevant features is then difficult. Also, training databases are not fully labeled. In speech and handwriting recognition for instance, data are labeled, at best, at the unit (phoneme or letter) level while it is often desirable to use a number of states for each unit, or even a number of modalities for a same unit (e.g. allograph in handwriting recognition).

This paper investigates the use of CRF models for such more general signal classification and segmentation tasks. We first introduce CRF and an extension called segmental CRF. Then we describe how to use CRF for dealing with multimodal classes and signal data and discuss corresponding inference and training algorithms. At last, we report experimental results concerning the eye movement challenge.

2 Conditional Random Fields for sequential data

Sequence labeling consists in identifying the sequence of labels $Y = y_1,...,y_T$ that best matches a sequence of observations $X = x_1...x_T$: $Y^* = \arg\max_Y P(Y/X)$. CRF are a particular instance of random fields. Figure 1 illustrates the difference between traditional HMM models and CRF. HMM (Fig. 1-a) are directed models where independence assumptions between two variables are expressed by the absence of edges. *CRF* are undirected graphical models, Figure 1-b shows a CRF with a chain structure. One must notice that CRF being conditional models, node *X* is observed so that *X* has not to be modeled. Hence CRF do not require any assumption about the distribution of *X*. From the random field theory, on can show [2] that the likelihood $P(Y/X)$ may be parameterized as:

$$P(Y/X,W) = \frac{e^{W.F(X,Y)}}{\sum_{Y'} e^{W.F(X,Y')}} = \frac{e^{W.F(X,Y)}}{Z_W(X)} \quad (1)$$

Where $Z_W(X) = \sum_{Y'} e^{W.F(X,Y')}$ is a normalization factor, $F(X,Y)$ is a feature vector and *W* is a weight vector. Features *F(X,Y)* are computed on maximal cliques of the graph. In the case of a chain structure (Fig. 1-b), these cliques are edges and vertices (i.e. a vertice y_t or an edge (y_{t-1}, y_t)).

In some cases there is a need to relax the Markovian hypothesis by allowing the process not to be Markovian within a state. [3] proposed for this semi-Markov CRF (SCRF). The main idea of these models is to use segmental features, computed on a segment of observations associated to a same label (i.e. node). Consider a segmentation of an input sequence $X = x_1, x_2,...,x_T$, this segmentation may be described as a sequence of segments $S = s_1, s_2,...,s_J$, with $J \le T$ and $s_j = (e_j, l_j, y_j)$

where e_j (l_j) stands for the entering (leaving) time in state (i.e. label) y_j. Segmental features are computed over segments of observations $x_{e_j},...,x_{l_j}$ corresponding to a particular label y_j. SCRF aims at computing $P(S/X)$ defined as in Eq. (1). To enable efficient dynamic programming, one assumes that the features can be expressed in terms of functions of X, s_j and y_{j-1}, these are segmental features:

$$F(X,S) = \sum_{j=1}^{|S|} F(X, y_{j-1}, s_j) = \sum_{j=1}^{|S|} F(X, y_j, y_{j-1}, e_j, l_j) \quad (2)$$

Inference in CRF and SCRF is performed with dynamic programming like algorithm. Depending on the underlying structure (chain, tree, or anything else) one can use Viterbi, Belief Propagation [4] or Loopy Belief Propagation [5]. Training in CRF consists in maximizing the log-likelihood *L(W)* based on a fully labeled database of *K* samples, $BA = \{(X_k, Y_k)\}_{k=1}^{K}$, where X_k is a sequence of observations and Y_k is the corresponding sequence of labels.

$$L(W) = \sum_{k=1}^{K} \log P(Y_k / X_k, W) = \sum_{k=1}^{N} ((W.F(X_k, Y_k) - \log Z_W(X_k)) \quad (3)$$

This convex criterion may be optimized using gradient ascent methods. Note that computing $Z_W(X)$ includes a summation over an exponential number of label sequences that may be computed efficiently using dynamic programming. Training SCRF is very similar to CRF training and also relies on a fully labeled database.

(a) (b)

Figure 1: Dynamic representation of HMM (a) and CRF (b) as graphical models, where grey nodes represent observed variables.

3 Semi-Markov CRF for signal segmentation

We investigate the use of segmental CRF for signal segmentation. When dealing with real signals, one has to consider the continuous nature of input data, the multimodality of the classes and one has to develop algorithms for learning models without a fully labeled dataset. Hence, in the following we will consider that, during training, the label Y_k corresponding to input sequence X_k consists in the sequence of classes in X_k, whatever the length of the segments associated to these classes.

To take into account multimodality (e.g. a letter may be written with different styles) we investigate the use of a few states in a Segmental CRF model for each class, each one corresponds to a modality of the class. We will note *K* the number of states sharing the same label. Since there are several states corresponding to the same label, there are a number of segmentations *S* that correspond to a particular label sequence *Y*. Following [6] we introduce hidden variables for multimodality and segmentation information and build upon their work to develop inference and training algorithms with incomplete data. Hence, when conditioned on an input *X* the likelihood of a label sequence *Y* is defined as:

$$P(Y/X) = \sum_{S \in S(Y)} P(S/X) = \frac{\sum_{S \in S(Y)} \sum_{M} e^{W.F(X,S,M)}}{\sum_{S'} \sum_{M'} e^{W.F(X,S',M')}} \quad (4)$$

Where *S(Y)* stands for the set of segmentations *S* (defined as in §2) corresponding to the sequence of labels *Y*, *M* denotes a sequence of hidden variables, with $m_i \in \{1,...,K\}$. The use of hidden variables *(S,M)* makes inference expensive:

$$Y^* = \arg\max_Y P(Y/X) = \arg\max_Y \left[\sum_{S \in S(Y),M} e^{W.F(X,S,M)} \right] \quad (5)$$

Where *Y*, *S* and *M* have the same length, say *T*. This expression cannot be computed with a dynamic programming routine since the maximum and sum operators cannot be exchanged. However, if one uses a Viterbi approximation where summation is replaced with the maximum operator, and one assumes that $e^{W.F(X,S,M)}$ may be factorized in a product of *T* independent terms then the double maximization may be computed efficiently. Hence we use:

$$Y^* \approx \arg\max_Y \left[\operatorname{Max}_{S \in S(Y),M} P(S,M/X,W) \right] \quad (6)$$

Training aims at maximizing the log-likelihood *L(W)*. Using Eq. (4), the derivative of the likelihood of the k^{th} training example is computed as:

$$\frac{\delta L_k(W)}{\delta w_v} = \sum_{S \in S(Y_k)} \sum_{M} P(S,M/Y_k,X_k,W).F_v(X_k,S,M) - \sum_{S'} \sum_{M'} P(S',M'/X_k,W).F_v(X_k,S',M') \quad (7)$$

This criterion is expressed in terms of expected values of the features under the current weight vector that are $E_{P(S,M/Y_k,X_k,W)} F_v(X_k,S,M)$ and $E_{P(S,M/X_k,W)} F_v(X_k,S,M)$. These terms may be calculated using a forward-backward like algorithm since the CRF is assumed to have a chain structure. Based on the chain structure of the models we used two types of features: local features (computed on vertices) $F^1(X,y_t,q_t,m_t)$ and transition features computed on edges, $F^2(X,y_t,y_{t-1},q_t,q_{t-1},m_t,m_{t-1})$.

4 Eye movement challenge data

Here is a quick description of the challenge and of the data for the competition 1 of the challenge, see [7] for more details. The eye movement challenge concerns implicit feedback for information retrieval. The experimental setup is as follows. A subject was first shown a question, and then a list of ten sentences (called titles), one of which contained the correct answer (C). Five of the sentences were known to be irrelevant (I), and four relevant for the question (R). The subject was instructed to identify the correct answer and then press 'enter' (which ended the eye movement measurement) and then type in the associated number in the next screen. There are 50 such assignments, shown to 11 subjects. The assignments were in Finnish, the mother tongue of the subjects. The objective of the challenge is, for a given assignment, to predict the correct classification labels (I, R, C) of the ten sentences (actually only those that have been viewed by the user) based on the eye movements

alone. The database is divided in a training set of 336 assignments and a test set (the validation set according to challenge terminology) of 149 assignments. The data of an assignment is in the form of a time series of 22-dimensional eye movement features derived from the ones generally used in eye movement research (see [7]), such as fixation duration, pupils diameter etc. It must be noticed that there is a 23^{th} feature that consists in the number of the title being viewed (between 1 and 10).

5 Experiments

We applied segmental CRF as those described in §3 to eye movement data. The aim is to label the ten titles with their correct labels (I, R, C). This may be done though segmenting an input sequence with a CRF whose states correspond to labels I, R and C. We investigated a number of models for this. All models have been trained with a regularized likelihood criterion in order to avoid over fitting [6]. These models work on vectors of segmental features computed over segments of observations. A simple way to define segmental feature vector would be the average feature vector over the observations of the segment. However, the average operator is not necessarily relevant. We used ideas in [7] to choose the most adequate aggregation operator (sum, mean or max) for each of the 22 features.

The first model is a simple one. It is a SCRF model with three nodes, one for class *R*, one for class *C* and one for class *I*. It works on segmental features where segments correspond to sequences where the user visits one particular title. There is no transition features, corresponding to the change from one title to another one. This model is called 3NL for 3 Nodes CRF with Local features only (no transition features) and 3NLT if transitions features are added. It must be noticed that since a title may be visited more than on time in an assignment it is desirable that the labeling algorithm be consistent, i.e. finds a unique label for every title. This is ensured, whatever the model used, by adding constraints in the decoding algorithm.

One can design more complicated models by distinguishing between the different visits of a same title. For example, one can imagine that a user who visits a title a second or a third time will not behave as he did the first time. Maybe he may take more time or quickly scan all the words in the title… Hence, we investigated the use of SCRF models with two or three states per class (I, R, C). In the two states models, a first state is dedicated to the first visit to a title of class R, C or I. The second state is dedicated to all other posterior visits to this title. When using 3 states per class, we distinguish among the first visit, the last visit and intermediate visits to a title. These models are named 6NL and 9NL depending on their number of states per class (2 or 3) if they make use of local features, and 6NLT and 9NLT if they make use of local and transition features.

Finally, we investigate the use of multimodal models. Going back to the first model 3NL, we consider the use of a few states per class, this time corresponding to different ways of visiting a title (there is no chronological constraints). Models are named 3N2ML for 3 states, 2 Modes per class, and Local features.

Table 1 reports experimental results for various SCRF-based models and for 4 additional systems. The first one is a benchmark HMM system. It works on the same input representation (feature vectors) and has the same number of states as there are nodes in the 6NL model. The three other systems are combination systems combine three classifiers votes.

A first comment about the results is that all SCRF models outperform the HMM system. Also, using more complex models is not systematically better useful. We investigated two ways for this, firstly by taking into account the number of the visit (increasing the number of states), secondly by taking into account multimodality

(increasing the number of modes). Using a few states per class in order to take into account the number of a visit of a title allows reaching up to 73% (9NL) while allowing multimodality leads to poorer results. Also, we did not succeed in using efficiently transition features, this is still under investigation. At last, voting systems did not improve much over singles classifiers although HMM and SCRF systems tend to be complimentary. There is certainly some room for improvements here. Note however that we observed more stability in the results of voting systems when training and testing on various parts of the database.

Table 1 – Comparison of various systems on the eye movement challenge task.

Technique	System's name	#states- #modes	Features	Accuracy (%)
SCRF	3NL	3 - 1	L	71
-	3N2ML	3 - 2	L	71.5
-	3NLT	3 - 1	L + T	68.9
-	6NL	6 - 1	L	71.8
-	9NL	9 - 1	L	73.2
-	9N2ML	9 - 2	L	70.8
-	9NLT	9 - 1	L + T	69.4
HMM	HMM	6 states	L	66.2
Combination of 3NL, 6NL, 9NL			L	72.1
Combination of HMM, 6NL, 9NL			L	72.3
Combination of HMM, 3NL, 6NL			L	71.9

6 Conclusion

We presented systems based on conditional random fields for signal classification and segmentation. In order to process signals such as eye movement, speech or handwriting, we investigated the use of segmental conditional random fields and introduced the use of hidden variables in order to handle partially labeled data and multimodal classes. Experimental results on the eye movement challenge data show that our CRF models outperform HMM, but all results are rather close showing the difficulty of the task.

References

[1] McCallum, A., Freitag, D., and Pereira, F. (2000) Maximum entropy Markov models for information extraction and segmentation. *In Proc. ICML.*

[2] Lafferty, J., McCallum, A., and Pereira, F. (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conf. on Machine Learning*, 282–289. Morgan Kaufmann, San Francisco, CA.

[3] Sarawagi, S., and Cohen, W. (2004) Semi-Markov Conditional Random Fields for Information Extraction. *Advances in Neural Information Processing Systems*.

[4] Weiss, Y. (2001) Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173-2200.

[5] Murphy, K., Weiss, Y., and Jordan, M. (1999) Loopy belief propagation for approximate inference: an empirical study. *In Proc. of the Conf. on Uncertainty in AI.*

[6] Quattoni A., Collins M. and Darrel T. (2004). Conditional Random Fields for Object Recognition. In Advances in Neural Information Processing Systems 17.

[7] Salojärvi, J., Puolamäki, K., Simola, J., Kovanen, L., Kojo, I., Kaski, S. (2005) Inferring Relevance from Eye Movements: Feature Extraction. Helsinki University of Technology, *Publications in Computer and Information Science*, Report A82.

Predicting Text Relevance from Sequential Reading Behavior

Michael Pfeiffer[*], **Amir R. Saffari A. A.**[*], **and Andreas Juffinger**[*]
Institute for Theoretical Computer Science
Graz University of Technology
8010 Graz, Austria
{pfeiffer, saffari, juffinger}@igi.tugraz.at

Abstract

In this paper we show that it is possible to make good predictions of text relevance, from only features of conscious eye movements during reading. We pay special attention to the order in which the lines of text are read, and compute simple features of this sequence. Artificial neural networks are applied to classify the relevance of the read lines. The use of ensemble techniques creates stable predictions and good generalization abilities. Using these methods we won the first competition of the PASCAL Inferring Relevance from Eye Movement Challenge [1].

1 Introduction

The objective of the PASCAL Inferring Relevance from Eye Movement Challenge [1] was to predict relevance of lines of text from eye movements of readers. The subjects were first shown a question and then a list of ten possible answers on a computer screen. The subject had to find the correct answer, then press 'Enter' and finally type in the chosen line number. Among the non-correct lines, there were four sentences which were relevant for the question, five were irrelevant. The task in the challenge is to identify not only the correct, but also the relevant lines of text, being provided only measurements of the eye movements of the subject. The gaze location on the screen and the pupil diameter were tracked during each assignment. For the first competition, the organizers had already segmented the trajectory data from the eye tracker and assigned fixations and saccades to the corresponding words and lines. 22 features that are commonly used in psychological eye movement studies were computed for every word. This made competition one a pure classification problem. In the second competition only the raw eye movement data and word coordinates were available. Thus preprocessing by segmentation and feature extraction was a main part of the problem. Our work focuses on competition one.

Eye movements during reading have been extensively studied in the psychological literature [2]. Different models pay more attention to either higher-level processes governing the reading behavior, or unconscious eye movements. In our approach, however we more or less neglected unconscious information, and computed features only from the sequence, in

[*] All authors made the same contribution to this project.

which the lines were read. Statistical analysis showed that for this task our simple features contained enough information for standard classifiers to obtain good results.

2 Feature Extraction

2.1 Features for Competition One

The organizers already provided a dataset with pre-computed features which have shown to be useful in other eye movement studies. Our approach, however, was quite different, since we wanted to find out as much as possible about the relevance of the sentences, only from the order in which the lines were read. This would imply that a much smaller fraction of data, namely only the line number, would have to be measured in order to distinguish between correct, relevant and irrelevant lines. Our features were computed for every seen line in an assignment, and classification was then done on this reduced feature set.

We found out from analyzing individual assignments, that there are some reoccurring patterns of sequential reading behavior. Due to the nature of the task, it was obvious that most readers usually look at the line containing the correct answer before pressing 'Enter' and thereby finishing his task. A heuristic rule which assigns the label *correct* to the last read line in each assignment, identifies 93.75% of all correct answers in the training set. This rule is facilitated by the fact that all assignments with incorrect answers have been filtered out by the organizers.

Additionally we noticed that the subjects spent significantly more time reading correct or relevant sentences than irrelevant ones. So we simply counted the number of measurements for each sentence, which turned out to be a very useful feature for classification. Secondly, the subjects jumped between lines of text while they are thinking about the correct answer. When they are deciding between different possibilities for a correct answer, it seems natural that their focus jumps between the lines within their consideration. Another assumption is that relevant sentences would rather be considered as a possible answer than an irrelevant sentence. We therefore counted, how often the subjects returned to a sentence they had already read before. This gave us a good estimation of the relevance that was attributed to this sentence by the subject, since the number of returns is significantly higher for correct or relevant lines. We also calculated jumps between the presumably correct line and all other sentences as another feature, but this did not improve the performance.

Most subjects showed a stereotypical reading behavior, i.e. they started with line number 1 and continued to read subsequent sentences, eventually jumping back to known answers. Therefore the position of a sentence within the list also plays a role, since some lines are almost always seen (like line 1), and some are more often skipped.

A problem we had to deal with was that not all of the possible answers in an assignment were read. Actually only in 40.8% of all assignments in the training set there was at least one measurement for every sentence. The subject was only instructed to look for the correct answer, but he was not told that the remaining lines contain either relevant or irrelevant sentences. Therefore it happened that the subject found the correct answer very early and did not even read a single relevant sentence. In this case it is of course almost impossible to make a distinction between relevant and irrelevant sentences. So we included the number of read lines as valuable context information, and also included a binary flag if all possible answers were read.

Since subjects tended to stop reading after they found the correct answer, we calculated the number of different sentences that the subjects read after or before having read that line. It turned out that the subjects read significantly fewer lines after reading the correct answer. In combination with the line number this feature thus provides evidence whether a sentence is the correct answer or not. Table 1 summarizes the 8 features that were used:

Table 1: Definition of features used for classification for every sentence in an assignment. Features marked with * are constant for all sentences in an assignment.

#	Feature	Description
1	`lineNr`	Line number within the 10 possible answers
2	`isLastLine`	1 if this sentence was the last one read during this assignment
3	`count`	Total number of measurements for this sentence
4	`returns`	Number of jumps to this sentence after first reading
5	`nrReadLines*`	Number of lines read during this assignment
6	`readAllLines*`	1 if all possible lines were read
7	`newLinesBefore`	Number of different sentences that the subject started to read *before* reading this line
8	`newLinesAfter`	Number of different sentences that the subject started to read *after* reading this line

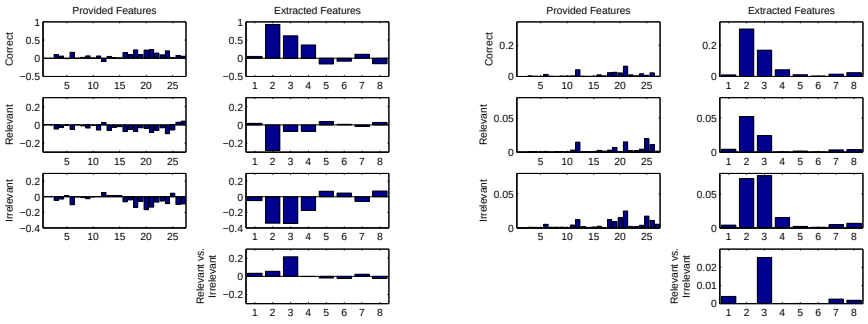
2.2 Statistical Analysis

In this section we analyze the statistical properties of our extracted features on the training set. We use mutual information (MI), a measure for arbitrary dependency between random variables, and linear correlation (LC) to compare the calculated features against the provided features for competition one. Mutual information has been used extensively in the literature for feature selection [4] because mutual information is invariant under linear transformations and takes into account the entire dependency structure of the random variables. On the other hand linear correlation is a natural measure for variable dependency. These measures are therefore good estimators for the relevance of features, although they do not take into account correlations among different features.

Firstly, we calculated the linear correlation and mutual information for each feature provided in the challenge with the class labels. We found that the original features are poorly correlated, as can be seen in the left columns of Figures 1 (a) and (b) respectively. In the first row of Figure 1(a) the LC and in 1(b) the MI, is shown for the correct vs. non-correct labels. The second rows show relevant vs. non-relevant and the third rows irrelevant vs. non-irrelevant. The last rows display LC and MI for the relevant vs. irrelevant problem, where correct lines have been removed. The provided features in general exhibit small MI values, except for features 12 (`lastSaccLen`), 21 (`regressDur`), and 25 (`nWordsInTitle`). The higher information of feature 12 about the correct labels confirmed the earlier statement about jumps between the correct answer and other titles. The high MI value for feature 21 is based on the same concept of going back for regression and re-reading. The duration of a regression is therefore more significant for classification than all other features.

Secondly, the mutual information and linear correlation for each of our extracted features were computed (see Figure 1 - right columns). The results confirm that most of these features are significantly more correlated with the class labels than the original features. Note that feature 21 has the highest MI value with 0.072, which is four times smaller than the maximum value for the new features (feature 2, with a MI value of 0.3042).

Some of the features exhibit neither high correlation nor high mutual information with the class labels. Nevertheless their inclusion boosted the performance of the classifiers. To explain this effect we calculated the MI of pairs of features with the class labels. We discovered that all features except number 2 (`isLastLine`, which is a good predictor on its own) show higher MI values in combination with each other than the sum of their



(a) Linear Correlation (b) Mutual Information

Figure 1: Correlation and mutual information analysis of different feature sets.

individual MI values. This means that even though some features are weak individual predictors, together they form a strong feature set.

We also calculated the MI and LC for our feature set after removing the correct titles to identify the features which are most significant for the relevant vs. irrelevant problem (see Figure 1 - last row). Feature 2 was almost constant for this dataset and therefore was not used for identifying relevant or irrelevant lines. Including all remaining features, even if their correlation and mutual information values were low, resulted in the best performance. Combined MI analysis again explained this effect.

3 Classification and Results

In this section we present our strategies and methods for building proper classifiers for the challenge datasets. Since the task is a 3-class classification problem, there exist two main strategies to solve the problem: the first one is to build a multi-class classifier, which gets the input pattern and assigns it to one of 3 classes. The second method, that we used in our systems, is to use a hierarchical classifier that first checks if the input pattern is a member of one of preselected classes (correct lines in this challenge) or not. If the result is negative, it passes the pattern to another classifier that assigns it to one of the other two remaining classes (relevant vs. irrelevant). In other words this technique decomposes a multi-class problem to a series of two-class pattern recognition problems.

Table 2: Correct detection rate (in %) of different 3-class classifiers.

	C4.5	**AdaBoost**	**SVM**	**MLP**
Train	68.17	68.62	63.85	66.71
Validation	69.19	65.24	66.23	71.09

We used the WEKA 3.4 package [3], which allowed us to quickly compare a variety of pattern recognition algorithm for this task. After manual tuning of the hyperparameters we calculated detection rates for different learning algorithms for training and validation sets averaged over 10 runs. The results are summarized in Table 2, showing an advantage for Multi-Layer Perceptrons (MLP) on the validation set. Note that these values are for 3-class classifiers, but we tried the same experiments for the two stage classification strategy mentioned above and the results were very similar.

3.1 Correct Line Identification

As can be seen in Figure 1, linear correlation and mutual information analysis shows that the features `isLastLine`, `count`, and `returns` are the most informative ones for identification of correct lines. Because of this we used only these features as input to our classifiers. We also changed the target labels to +1 for correct lines and -1 for the rest (relevant and irrelevant).

For the rest of the project we switched to a more efficient and flexible tool for training neural networks, namely the MATLAB Neural Networks Toolbox. A MLP with 3 hidden neurons and hyperbolic tangent activation functions was trained with scaled conjugate gradient backpropagation. We tried different numbers of hidden neurons, but since the performance did not change significantly with more neurons, we chose the simplest network to avoid overfitting. In addition the error on the validation set was used as stopping criterion for training. The overall performance on training, validation, and test sets are shown in Table 3. Ensemble methods, as discussed in the next section, were also tried out for this task, but since the recognition rate was almost constant for different MLPs, we decided to use a single classifier instead.

3.2 Relevant vs. Irrelevant Lines Identification

For this task we first removed the predicted correct lines from the previous classifier for training, validation, and test sets. Then the feature `isLastLine` was removed, since correlation and mutual information analysis showed that it had no major contribution to this task. We also changed the target labels to +1 for relevant and -1 for irrelevant lines. As another preprocessing step, we normalized the feature values to have zero mean and unit variance according to the combined training, validation and test sets.

Different MLPs were trained, and for each network the numbers of hidden neurons was randomly selected between 4 and 10. The activation function was hyperbolic tangent for all neurons and we trained our networks using the Levenberg-Marquardt backpropagation algorithm of MATLAB's Neural Networks Toolbox. As before we used the error on the validation set as stopping criterion.

Table 3: Correct detection rate (in %) for two stage classifiers. Row 2 shows the average performance of single MLPs, rows 3 and 4 correspond to the two different ensemble methods for relevant vs. irrelevant line classification. Overall accuracy for the 3-class predictions is shown in parentheses.

	Train	**Validation**	**Test**
Correct vs. Rest	98.54	98.52	98.72
Single MLP	63.01 (67.26)	66.57 (69.81)	67.91 (71.03)
Best-Of Ensemble	64.21 (68.02)	68.01 (71.25)	69.26 (72.31)
Outlier-Filtered Ensemble	64.25 (68.06)	67.73 (71.17)	69.09 (72.16)

The main difference compared to the previous setup for correct line identification was that we used an ensemble averaging method to improve and stabilize our recognition rate and avoid possible overfitting. It has been shown that in most cases ensemble averaging methods improve the generalization properties of classifiers [5, 6]. So we averaged the confidence values (outputs of the networks) over all ensemble members and then used it as a decision criterion. We used two different methods to select ensemble members: one method, named *Best-Of* ensemble, selected the best 10 networks out of 15 different trained networks. The other approach, named *Outlier-Filtered* ensemble, filtered out networks that showed relatively high error rates. 5 networks were selected, and the selection threshold

was set at an error rate of 38%. For both methods we used the error on the validation set as our selection criterion.

The overall results are given in rows 2-4 of Table 3 for training, validation and test sets. We first show the average performance of single MLPs, and then the accuracy for both ensemble selection methods in the last two rows. The benefits of using ensemble methods can be seen by comparing row 2 with rows 3 and 4, since the error on all sets was on average reduced by more than 1% (for both methods). In addition the variance of the classification error was also significantly reduced. In competition one, *Best-Of* ensemble finished first, *Outlier-Filtered* ensemble finished second. The next best result was 0.9% lower than our best performance.

Furthermore, we tried a post-processing step in which the main goal was to correct inconsistent decisions such as having more than 4 relevant or more than 5 irrelevant detections. The confidence values of the ensemble were used as the basis for the post-processing decision. We tried to change the labels of less probable excessive detections to the other class. The major problem was that in most cases the confidence value was not a good representative of being a member of a class when there were excessive detections. So we decided not to use this post-processing method for our classifiers.

4 Conclusion

Our feature extraction and classification approaches were highly successful in this challenge. The ensemble methods proved to be very stable and exhibited very good generalization performance. In addition we showed that a lot of information about the relevance of read lines can be extracted from features about sequential reading behavior. We do not claim, however that unconscious eye movements during reading are not informative for this task, but our results show that reasonable accuracy can be obtained without them.

Acknowledgments

This work was supported in part by PASCAL Network of Excellence, IST-2002-506778, the Austrian Science Fund FWF, project number S9102-N04, and the MISTRAL-project financed by the Austrian Research Promotion Agency, project contract number 809264/9338. This publication only reflects the authors' views.

References

[1] Salojärvi, J., Puolamäki K., Simola J., Kovanen L., Kojo I. & Kaski S. (2005) Inferring Relevance from Eye Movements: Feature Extraction. *Publications in Computer and Information Science*, Report A82. Helsinki University of Technology.

[2] Rayner K. (1998) Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, **124**, pp. 372-422.

[3] Witten I.H & Frank E. (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.

[4] Blum, A. & Langley, P. (1997) Selection of relevant features and examples in machine learning. *Artificial Intelligence*, **97**(1-2), pp. 245-271.

[5] Opitz D. & Maclin R. (1999) Popular ensemble methods: an empirical study, *Journal of Artificial Intelligence Research*, **11**, pp. 169-198.

[6] Bauer E. & Kohavi R. (1999) An empirical comparison of voting classifi cation algorithms: bagging, boosting, and variants. *Machine Learning*, **36**, pp. 105-142.

Inferring Relevance from Eye Movements Using Generic Neural Microcircuits

Tuomas Lepola
Department of Computer Science, University of Helsinki
P.O. Box 68, FIN-00014, Finland
Tuomas.Lepola@cs.helsinki.fi

Abstract

In the "Inferring Relevance from Eye Movements Challenge 2005", contestants were asked to apply machine learning techniques for predicting sentence relevancies based on eye movements of the readers. The two-part competition consisted of a classification problem and a time series analysis problem. In our winning solution to the time series analysis problem, we applied generic neural microcircuit as a nonlinear operator on time series with discriminative classifier as a readout. The results suggest that the model used has some practical merits as a generic time series analysis tool.

1 Introduction

The "Inferring Relevance from Eye Movements Challenge 2005" [1] was organized in the form of a two-part data analysis competition. The contestant were given two competition data sets collected from the experimental setting where subjects were asked to identify correct answers to each assignment presented. Each of these assignments consisted of a question and 10 sentences of which five were irrelevant to the question, and of the remaining five relevant sentences one was the correct answer. In the test setting, the sentences were presented on a computer display and the eye movements and the pupil diameters of the subjects were recorded and stored as a form of time series. The contestants were asked to identify, on the basis of the eye movements, which sentences of each assignment were irrelevant, relevant, and correct. In Competition 1 the time series data were preprocessed into a form of traditional classification data. This preprocessing method segments the time series data in a manner typical to psychological research of reading. In contrast to this, in Competition 2, only the raw time series data was presented to contestants.

There were three main difficulties in the competition setting. Firstly, according to the description of the test setup, "the subject was instructed to identify the correct answer" [1]. In particular, the subjects were not instructed to read through all sentences. In this sense, there was a slight discrepancy between the experimental setting and the challenge. This discrepancy creates the difficulty of how to classify unseen sentences, since many subjects read efficiently and completely skipped the remaining sentences after finding the correct answer. These unseen sentences were ignored while computing the prediction accuracy in Competition 1. However, in Competition 2 all sentences affected the accuracy score. Secondly, the irrelevant sentences may have contained some words which drew the attention of

the subject in spite of the sentence not being relevant in the assignment. Hence the baseline methods applied by the organizers indicated that the separation of irrelevant and relevant sentences was the most difficult part of the prediction task. Thirdly, the eye-movement data is inherently noisy which in general makes data analysis always more challenging.

The winners of both competitions were selected based on prediction accuracy for the test sets. In the following, we present the winning solution in Competition 2: the model, the implementation and the results, and our conclusions based on the results.

2 The Model

The generic neural microcircuit model has been introduced recently as a realistic model of cortical columns (see [2] for a comprehensive introduction). The model encapsulates the generic and the stereotypical characteristics of cortical columns in a useful theoretical and practical framework. The generic neural microcircuit implements a nonlinear operator L^M which transforms the input time series to the dynamic state $x^M(t)$ of the circuit at time t. As detailed in [2], the essential property of this operator is the pointwise separation property which informally means that different input signals lead to robustly separated dynamic states. The dynamic state $x^M(t)$ of the circuit is defined as the output of all the neurons in the circuit at time t. The readout of the circuit implements a memoryless map of the circuit state to the output time series. Putting all this together, we get the output of the circuit, given input $x(\cdot)$ at time t, to be

$$y(t) = f^M((L^M x)(t)),$$

where f^M is the readout map. It has been shown in [2] that if the readout map f^M has a certain approximation property, and assuming the pointwise separation property for L^M, then the whole circuit has "universal power for computation with fading memory on functions of time". More specifically, the proof technique using the Stone-Weierstrass theorem also applied in [3] shows that any time-invariant operator having fading memory can be approximated arbitrary closely by this circuit. Thus this model has the necessary flexibility to many time series processing applications. Our experiments with the eye movement time series data supports also the practical applicability of the model.

We implemented a generic neural microcircuit as a three-dimensional lattice of integrate-and-fire type spiking neurons as in [2]. Each circuit comprises a neural column of 135 neurons (dimensions 3 x 3 x 15). The connectivity structure of the network is highly recurrent and it is governed by a distribution which, roughly, renders connections (excitatory or inhibitatory) more likely between neurons close by than neurons far apart. The synaptic connections are using dynamical synapses (see details of the model further on). As shown in [2], the separation property is enhanced by combining parallel columns as a one circuit. Hence as whole we applied four parallel circuits in Competition 2 instead of the one used in Competition 1.

The neural column was constructed as follows. The membrane potential u_i of each neuron i is given by the standard integrate-and-fire model with the membrane time constant τ_m and the total membrane resistance R,

$$\tau_m \frac{du_i}{dt} = -u_i(t) + R \sum_j w_{ij}(t) \sum_{f \in \mathcal{F}_j} \alpha_j(t - t_j^{(f)}), \tag{1}$$

where the postsynaptic input current was modeled using

$$\alpha_j(s) = \frac{s - \Delta_j^{ax}}{\tau_s^2} \exp(-\frac{s - \Delta_j^{ax}}{\tau_s}) \Theta(s - \Delta_j^{ax}).$$

The parameter Δ_j^{ax} specifies axonal transmission delay, τ_s provides synaptic time constant, and Θ is the Heaviside function (see [4] for details). Additionally, the model was

augmented by the standard firing threshold rule and the absolute refractory period. The threshold criterion specifies the set of spikes $\mathcal{F}_j$ and the firing times $t^{(f)}$ of Equation (1).

The synaptic efficacy function w_{ij} was modeled according to the phenomenological model of frequency-dependent synaptic dynamics given in [5]. This model formulates synaptic facilitation and depression as a function of the absolute synaptic efficacy and the fraction of available and unavailable synaptic efficacy. See [5] for details of this intricate model. The parameter values of the whole model were selected for the computer simulation as in Appendix B of [2]. These parameter values could be argued to be biologically reasonable and the whole model is characterized by reasonable biological realism with the eye on feasible computer simulation (see Section 3 for discussion on computational issues).

The dynamic state of the circuit was read out as a vector of spike trains from all neurons of the circuit. Furthermore, this vector was transformed to time varying output currents with the effect of each spike upon current decaying exponentially, and 20 ms time-window was applied to discretize this output signal. Hence, we obtained for each input signal and for each column a 135-dimensional discrete output time series.

It should be emphasized that no learning is involved in applying the generic neural microcircuit. The circuit could be the same for each time series given as an input. Only the readout map is selected according to the task, for example, as a linear classifier. This means that the learning task is easy compared to e.g. adjusting the parameters of a nonlinear recurrent neural network with supervised learning. In terms of computational power of the circuit, the readout map only needs to posses some weak capabilities mentioned above. Moreover, the readout can be memoryless, that is, each discrete sample of the circuit output could be classified independently of the past of the series. Hence the circuit provides a kind of natural preprocessing of a time series data to a classification data.

However, instead of using a linear perceptron network as a readout map like in [2], we applied discriminative classification to the circuit output. More specifically, our goal was to directly estimate the parameters of a distribution $P(Y|X)$ where Y is a binary variable specifying if the gaze of the subject is on relevant sentence or not given the output vector X (correct answers were considered as relevant, see the reasoning given in the next section). We assumed that $P(Y|X)$ could be learned reasonably well using logistic regression and proceeded to maximize the conditional data log likelihood of the weight vector W:

$$l(W) = -\sum_t (Y^t(w_0 + \sum_i w_i X_i^t) - \ln(1 + \exp(w_0 + \sum_i w_i X_i^t))),$$

where the superscript t denotes the variable at the time step t. We applied the standard conjugate gradient descent to optimize the weight vector. Due to the simplicity of the optimization task, the convergence to the global maximum is guaranteed. However, to avoid overfitting, we applied weight regularization to penalize weights too large and furthermore, we proceeded with data specific cross-validation and used the validation set (see next section) to ensure the generalization capability of our model.

Finally, we applied a Bayesian approach to compute, given the estimated distribution $P(Y|X)$, the probability that a sentence in an assignment is relevant. Due to nature of the Competition 2 data, more information was available concerning the eye movements of the subjects between fixations than in the Competition 1 data. The subjects may have visited some relevant sentences very briefly, and especially they have not fixated on some relevant sentences. Hence very little data was available of some sentences compared to others. We applied parameter smoothing (as in the m-estimate of probability, see e.g. [6]) with uniform prior to augment this weakness.

The reasoning behind the use of logistic regression as a readout processor is as follows. Firstly, by nature of the generic neural microcircuit, the nonlinear transformation removes the necessity of the readout to have memory. Hence we can classify the data as independent

samples from some unknown distribution. All information is available in the dynamic state of the circuit at some specified moment of time to approximate the output signal at that time. The key issue is to assess whether this approximation can be learned easily and reasonably well, and it was assumed that logistic regression performs in this respect at least as well as the linear parallel perceptron applied in [2]. Secondly, informal testing with an advanced naive Bayes classifier B-Course (available as an online service, see [7] for details) seemed to indicate that the performance of the naive Bayes classifier was seriously hurt by the obvious violation of the independence assumption of the variables. Although, more generally, the gaussian naive Bayes model and logistic regression are intimately related, logistic regression is not as tightly constrained by the conditional independence assumption as the gaussian naive Bayes model. It seems that our classification problem, where data is in abundance, is an example of a case where logistic regression outperforms the gaussian naive Bayes model [8]. Due to time limitations of the challenge, we did not analyze this observation further.

3 Implementation and Results

The raw challenge data of Competition 2, consisting of the original horizontal and vertical eye movements and pupil diameter data, was preprocessed to six time series. All time series were normalized and injected as an input current to randomly selected neurons of the circuit. We applied very simple transformation in the spirit of feature extraction described in [1]. At the level of words and sentences we recorded cumulative visits and revisits, local movements inside a word relative to optimal viewing position [9], the relative movements between the words inside a sentence, and finally, the relative movements between the sentences of an assignment. Additionally, we registered the pupil diameter changes. Since some pupil diameter readings were clearly anomalously large, we set all pupil diameter readings above 6 mm to the maximum of 6 mm.

The simulation of the generic neural microcircuit involved numerical integration. We applied the standard Runge-Kutta-Fehlberg method (4th order, 5th order error estimate) with adaptive stepsize and a modification to detect firing condition to simulate the dynamics of the generic neural microcircuit model. All simulation software were implemented using C++ programming language in the Linux computing environment. Since we were able to distribute the computation of separate columns and readouts to different computing nodes and processors, this greatly enhanced the throughput of the data processing. Typically a two-way Intel Xeon processor node with hyperthreading support provided throughput as high as 900 spikes per second and a two-way 64-bit AMD Opteron node even higher. Most of the preprocessing and postprocessing of the data and the results were run in the Matlab environment.

Since our method is clearly oriented to time series processing and we had very limited time for the competition, our emphasis was on Competition 2. However, a modest attempt was also made in Competition 1 since we realized that we could estimate the original time series from the classification data to some degree of accuracy. Obviously, in the training setting we could have used the Competition 2 training set to train our model, but this would have created a problem with the test set since the Competition 2 test set was made available after Competition 1 was closed. Instead we estimated the original time series based on only the Competition 1 data with three main estimation methods. Firstly, we estimated the word lengths of each assignment by averaging and using the knowledge of first and last fixations to the word, and the optimal viewing position. Secondly, we added gaussian noise to fixation positions to simulate the measurement errors since the average accuracy of equipment was given in [1]. Thirdly, we interpolated the eye movements between fixations using the facts known about eye movement speed function and the dimensions of the display and positioning of the subject in front of the display [1].

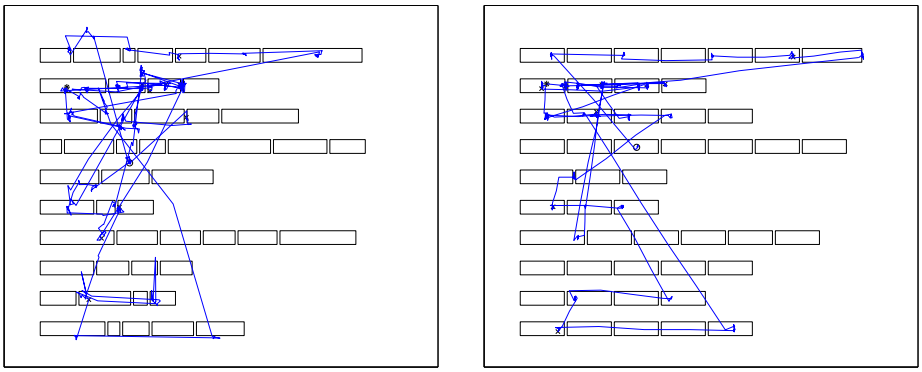


Figure 1: The actual eye trajectory and an estimation

Figure 1 shows one assignment from the training data. The actual eye movements are shown on the left and an estimation based on classification data is given on the right. As can be seen, qualitatively the eye movement trajectories look quite the same. However, some important trajectories are missing in the long gaps between some fixations. These short visits seemed to be crucial in separating some relevant and irrelevant sentences.

In Competition 1 we preprocessed the estimated time series as described above. The resulting six time series were given as an input to a single column of the generic neural microcircuit. The parameters of the readout map were estimated as specified in the previous section using the training data. After training with leave-one-assignment-out and leave-one-subject-out cross-validation, the model gave 60.9% accuracy with the validation data. Since the test data turned out to give the accuracy of 60.7%, we can conclude that this simplified method performed robustly, although poorly compared to other competitors. We argue that this is mainly due to methodological discrepancy between the time series oriented model and the classification data.

In Competition 2 we applied the full model to predict the relevant sentences. After the time series preprocessing performed as above, we used the four-column generic neural microcircuit model to predict the probabilities of the sentences being relevant. The training with conjugate gradient descent optimization and cross-validation was essentially the same as described above. The most relevant sentence of the assignment was chosen to be the correct answer. If, instead of binary classification, we did three class classification (correct, relevant, irrelevant) we noticed a degrading performance in separating the relevant sentences from irrelevant ones. We concluded that we should concentrate on solving this harder separation problem, since this was in the spirit of the challenge. The test results proved that this approach was successful and overfitting of the noisy data was avoided.

Table 1: Confusion matrices of the Competition 2 validation and test sets

Validation	$I(68.3\%)$	$R(56.7\%)$	$C(75.2\%)$
$I(68.3\%)$	509	226	10
$R(56.7\%)$	231	338	27
$C(75.2\%)$	5	32	112
Test	$I(68.7\%)$	$R(63.9\%)$	$C(77.8\%)$
$I(68.7\%)$	618	276	6
$R(56.4\%)$	282	406	32
$C(78.9\%)$	0	38	142

Table 1 summarizes the results of the method we applied on the Competition 2 validation and test data sets. As can be seen in the confusion matrices, the strength of the method was exactly in separating the irrelevant and relevant sentences. The accuracies were 64.4% on validation data and 64.8% on test data. This indicates that the method is characterized by robust prediction performance, even in the presence of the unseen sentences.

4 Conclusions

Our solution in the eye movement data analysis challenge was based on generic neural microcircuit model with discriminative classification through logistic regression. Our solution won Competition 2 which involved predicting sentence relevancies using the raw time series data. We conclude that the neural microcircuit used exhibited practical genericity in time series analysis, since only very simple preprocessing was needed for the task at hand. We emphasize that the learning problem is easy with this model, since the microcircuit is completely generic and a linear classifier is sufficient to extract useful predictions from the model. Additionally, the test results indicate that the model is robust and does not overfit easily although the time series data was noisy. Furthermore, the results suggest that future work on at least three different areas would be interesting. Firstly, in the context of this modeling problem, it would be important to find methods for improving the classification of correct sentences without introducing overfitting. Secondly, this data analysis problem seems to be a case where logistic regression outperforms gaussian naive Bayes model and it might be of value to do more experimentation with the challenge data to compare the practical performances of the two models. Thirdly, the spatial structure of the generic neural microcircuit might benefit from improvements in biological realism.

Acknowledgments

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the author's views.

References

[1] Salojärvi, J., Puolamäki, K., Simola, J., Kovanen, L., Kojo, I. & Kaski, S. (2005) Inferring relevance from eye movements: Feature extraction. *Helsinki University of Technology, Publications in Computer and Information Science, Report A82*.

[2] Maass, W., Natschläger, T. & Markram, H. (2002) Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation* **14**:2531-2560.

[3] Boyd, S. & Chua, L. O. (1985) Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems* **32**:1150-1161.

[4] Gestner, W. & Kistler, W. (2002) *Spiking Neuron Models.* Cambridge University Press.

[5] Markram, H., Wang, Y. & Tsodyks M. (1998) Differential signaling via the same axon of neocortical pyramidal neurons. *Proc. Natl. Acad. Sci.* **95**:5323-5328.

[6] Mitchell, T. M. (1997) *Machine Learning.* McGraw-Hill.

[7] Myllymäki, P., Silander, T., Tirri, H. & Uronen, P. (2002) B-Course: A web-based tool for Bayesian and causal data analysis. *International Journal on Artificial Intelligence Tools* **11**:369-387.

[8] Ng, A. Y. & Jordan, M. I. (2002) On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In T. G. Dietterich, S. Becker and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems 14*, MIT Press.

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

Predicting, analysing, and guiding eye movements

Michael Dorr[*]
Institute for Neuro- and Bioinformatics
University of Lübeck
Germany
dorr@inb.uni-luebeck.de

Martin Böhme
Institute for Neuro- and Bioinformatics
University of Lübeck
Germany
boehme@inb.uni-luebeck.de

Thomas Martinetz
Institute for Neuro- and Bioinformatics
University of Lübeck
Germany
martinetz@inb.uni-luebeck.de

Erhardt Barth
Institute for Neuro- and Bioinformatics
University of Lübeck
Germany
barth@inb.uni-luebeck.de

Abstract

In this paper, we will present an overview of our work that is aimed at integrating gaze into visual communication systems by measuring and guiding eye movements [1]. This requires investigating and modelling how eye movements are determined by the visual input, modelling what is relevant to the user, and new technological developments for better eye tracking and fast gaze-contingent graphics. A number of challenges remain, some of which may be solved by machine-learning techniques, e.g. predicting eye movements and inferring a person's intent.

1 Background

Vision is the dominant perceptual channel through which we interact with information and communication systems, but one major limitation of our visual communication capabilities is that we can attend to only a very limited number of features and events at any one time (e.g., [2]). This fact has severe consequences for visual communication, because what is effectively communicated depends to a large degree on those mechanisms in the brain that deploy our attentional resources and determine where we direct our eye movements, i.e. our gaze.

Therefore, future information and communication systems should use gaze guidance to help the users deploy their limited attentional resources more effectively. Gaze guidance here means that the user follows a prescribed pattern with their gaze, thus taking in information in a specific, potentially more efficient way.

When dealing with the problem of guiding a user's gaze, we face several challenges. In this paper, we will give an overview of the issues that we have begun to address so far, but there remain a number of open problems.

[*] http://www.inb.uni-luebeck.de

The first challenge is that we need to analyse in more detail how humans watch dynamic scenes. The majority of previous research on eye movements has dealt with static scenes only, mainly because of the technical problems inherent in recording eye movements on movies. However, we believe that it is more practicable to determine what drives eye movements in dynamic scenes. Bottom-up features, that is features that are directly computable from an image sequence such as brightness, colour, or motion, should have a greater influence on directing gaze here than in static scenes. Accordingly, attempts have recently been made at modelling what low-level features determine eye movements in moving scenes as well [3]. Based on these findings, we have been able, at least to some extent, to predict where observers will direct their gaze from a number of previously attended locations [4].

A further requirement is a quality function that estimates how well-suited an observer's gaze pattern is for a given image sequence. There are basically two approaches to obtain an optimal gaze pattern. It is well known that experts, for example experienced car drivers, employ viewing strategies different from those of novices. Thus, the gaze pattern of an expert could be recorded and "replayed" to the user. The more generic approach uses an image-processing algorithm that could identify the most informative regions in a scene. Of course, we also need a model of both the task at hand and the observer's intentions to decide which information might be relevant to the observer.

Finally, the eye movements actually need to be guided to follow the intended gaze pattern. In a strict sense, this will be impossible to achieve, as an observer might consciously choose to only focus on one specific aspect of a scene. Nevertheless, we believe that for most purposes, it will suffice to significantly increase the likelihood that certain locations will be fixated, while suppressing other potential saccade targets. Indeed, we have developed a number of spatio-temporal transformations that, as we were able to show, change the eye movements of observers, although the guidance still needs to become more specific.

When all these challenges have been met, we furthermore not only want to change the observer's eye movements, but also achieve a change in behaviour, i.e. an improvement in actual task performance. In the next section, we will give an overview of potential applications.

2 Applications

An important potential application of gaze-guidance systems is augmented vision. Augmented-vision systems can be designed to integrate human vision and computer vision. For example, in a car, the driver's attention can be directed towards a pedestrian who has been detected by sensors looking out of the car.

A further application is the use in training systems. It is known that experts, for example experienced pilots, scan their environment in a way that substantially differs from how inexperienced viewers would. We believe that by recording the gaze pattern of experts and applying it to novices, we can evoke a sub-conscious learning effect.

Finally, current technical visual communication systems are based on the physical properties of images and cannot improve the communication process as such, because they do not address the question of what message is conveyed by an image or a video. Future communication systems' images and movies can be defined not only by brightness and colour, but will also be augmented with a recommendation of where to look, of how to view the images. For this to succeed, such systems will also have to take into account the user's intentions.

3 The machine-learning perspective

The human visual system is highly complex. This complexity is increased even further when we extend our view to include the higher cognitive functions that also play a role in controlling the direction of gaze, such as alertness, emotional state, or intent. Therefore, we believe that it will be impossible to distill a set of fixed parameters that will allow gaze guidance to work in all situations, for all users. Rather, we believe that the gaze-guidance display will have to continuously adapt to the user and the task at hand.

In the gaze-guidance display, what is displayed is a function of the user's gaze, while at the same time the display influences the user's gaze. In this closed loop, there exists a multitude of parameters that need to be adjusted in an on-line fashion. Different users may have different physiological characteristics, such as saccadic latency, or different cognitive strategies, attentional states, or expectations. Lastly, the search space of spatio-temporal transformations that might possibly be used to guide gaze is too vast to be explored systematically. We have implemented some basic transformations (see section 4), but fine-tuning will have to be done in an unsupervised, continuously evolving manner.

The following sections will outline some of the issues we have addressed so far with machine-learning techniques and other methods.

3.1 Analysis of eye movements

We have investigated the variability of eye movements on dynamic natural scenes [5]. To this end, we collected a large data set of gaze samples from 54 subjects watching a variety of short video clips (20 s duration each). For each movie frame, clusters of gaze samples were extracted by an unsupervised machine-learning algorithm. First, a fixation map was created by a superposition of Gaussians centered at each gaze sample. From the resulting map, up to $n = 20$ maxima were extracted by iteratively applying a lateral inhibition scheme. Then, clusters were formed using a simple distance threshold. Results show that there exist "hot spots" which contain a high number of fixation locations. On average, 5-15 clusters (2-5% of the viewing area) account for 60% of all fixations (see Fig. 1 for an example).

3.2 Eye movement predictions

The model we use to predict where an observer is going to look is composed of two distinct parts. This separation is motivated by the two fundamental types of eye movements that are relevant to our purposes. First, saccades are ballistic high-velocity eye movements that serve to move the fovea from one fixation location to another; during a saccade, most of the visual input is suppressed so that, for example, we do not perceive the blur induced by the motion of the visual scenery across the retina. We define the task of saccade prediction as predicting the target of the saccade, not the complete saccade trajectory, because the latter is irrelevant for our purposes. The second type of eye movements comprises all movements that are made between saccades. These movements can be further classified into a variety of types, but for the present purpose, they share the common characteristic that velocity is relatively low. To model such intersaccadic eye movements, we use a supervised-learning technique that, from a history of previous gaze samples, predicts the gaze position in the next time step.

For the prediction of saccade targets, one ideally would be able to predict a single location that has a high probability of being the next saccade target. To achieve this, however, a complete understanding of the higher decision processes involved in saccadic programming would be required. For example, an observer might choose to fixate one part of a scene over another for purely semantic reasons. Therefore, we restrict ourselves to predicting only a certain number of locations that are likely to be fixated as the next saccade target. We have

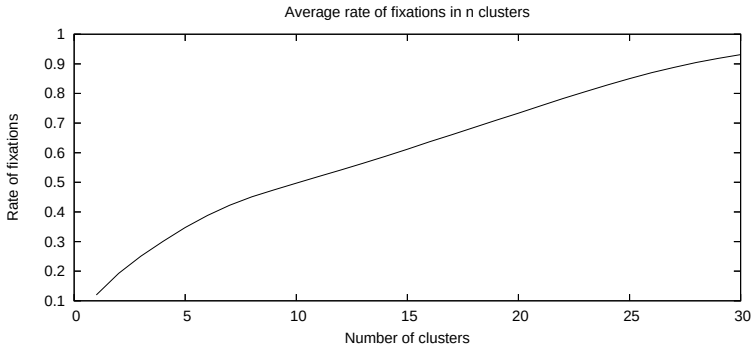


Figure 1: Rate of fixations that fall into the first n extracted clusters, for one example movie. These rates were computed on a per-frame basis and averaged across the whole movie.

attempted to use machine learning techniques to predict which of the candidate locations will be chosen, but with only limited results so far.

To extract candidate locations for saccade targets from a video, we use a saliency map that assigns a certain degree of saliency to every location in every frame of the video sequence. Various techniques exist for computing saliency maps, most of which are intended to model the processes in the human visual system that generate potential saccade targets [3].

The saliency map used here is based on the spatio-temporal curvature of the image sequence. The curvature is computed here using the determinant of the structure tensor, which is defined as the locally averaged outer product of the (x, y, t) intensity gradient. To avoid all candidate locations being extracted from only a single small high-curvature region in an image, we extract locations by iteratively applying a lateral inhibition algorithm, so that locations with a high saliency close to a local maximum become suppressed.

Fig. 2 compares the performance of our predictor with that of a predictor that uses an empirical saliency map, which is derived from the recorded eye movement data as described in section 3.1: Clusters with a high density of fixations are assigned a high saliency. This empirical saliency map gives an upper bound of what we can expect to achieve with a purely bottom-up approach, without modelling the user's top-down influences.

The results show that, currently, the performance of our predictor is about halfway between the results one would obtain by guessing locations at random and that of the ideal predictor based on the empirical saliency map. For a detailed discussion of our predictor, see [4].

4 Current state of the art

The final goal of our gaze guidance system is to direct the user's attention to a specific part of a scene without the user noticing this guidance. Apart from our work on modelling which image features attract gaze, we have therefore also conducted experiments with sev-

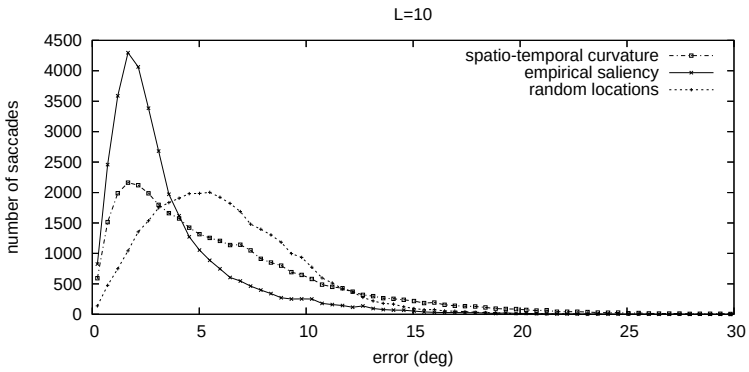


Figure 2: Error distributions for different predictors. The error is the distance of the recorded saccade target to the closest of $L = 10$ predicted candidate locations.

eral different spatio-temporal transformations designed to alter eye movement characteristics. These transformations were based on observations made with synthetic stimuli, which are commonly used in experiments that investigate attentional effects. The first set of transformations was motivated by the well-known fact that sudden object onsets in the visual periphery can attract attention. We chose to briefly superimpose small bright red dots on the movie. In about 50% of trials, saccades were initiated towards the location of the flashed red dot. Because the typical saccadic latency of about 200 ms exceeds the presentation time of the dot, which was set to 120 ms, the red dot was already switched off by the time the saccade was finished, so that in about 65% of cases, this stimulation remained invisible. Similar results were obtained in an experiment where the red dot was replaced by a looming stimulus. Nevertheless, the exact parameters for an optimal guidance effect, such as size, contrast, duration, or the timing with regard to previous saccades, still need to be determined, ideally by an automated learning process.

For a second, more complex set of transformations, we have developed a gaze-contingent display that can in real time change the spatio-temporal content of an image sequence as a function of where the observer is looking [6], based on earlier work that manipulated only spatial resolution [7]. For example, we can selectively blur high temporal frequencies in the visual periphery, which are known to evoke saccades. Because of the limited perceptual capabilities of the human visual system in the periphery, this blur remains unnoticed. Nonetheless, we were able to show that such peripheral temporal blur suppresses saccades towards the periphery. Next, we plan to specifically change the spatio-temporal content only at certain locations in an image.

5 Conclusion

We have here described the efforts we have made to not only infer and predict human behaviour (eye movements) but also change it such as to improve human performance. Preliminary results indicate that it should be possible to guide the gaze of a person [8]. A number of problems that need to be solved can be addressed by machine-learning techniques. The ultimate goal would be to find the optimal way to display information such as to minimize the error between the actual and the desired performance of a person performing certain actions in a particular environment, e.g. to avoid traffic accidents, or the difference between the information that is intended to be received and the one that is actually received, e.g. by a person watching a movie or a news programme.

Acknowledgments

Research is supported by the German Ministry of Education and Research (BMBF) under grant number 01IBC01, ModKog.

References

[1] Information technology for active perception homepage, 2001. http://www.inb.uni-luebeck.de/Itap/.

[2] J Kevin O'Regan, Ronald A Rensink, and James J Clark. Change-blindness as a result of 'mudsplashes'. *Nature*, 398:34, 1999.

[3] L Itti. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 2005.

[4] Martin Böhme, Michael Dorr, Christopher Krause, Thomas Martinetz, and Erhardt Barth. Eye Movement Predictions on Natural Videos. *Neurocomputing*, 2005. (in press).

[5] Michael Dorr, Martin Böhme, Jan Drewes, Karl R Gegenfurtner, and Erhardt Barth. Variability of eye movements on high-resolution natural videos. In Heinrich H Bülthoff, Hanspeter A Mallot, Rolf Ulrich, and Felix A Wichmann, editors, *Proceedings of the 8th Tübinger Perception Conference*, page 162, 2005.

[6] Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth. A gaze-contingent display with variable temporal resolution. In *Proceedings of the BIP Workshop on Bioinspired Information Processing, Lübeck, Germany*, page 18, 2005.

[7] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In B E Rogowitz and T N Pappas, editors, *Human Vision and Electronic Imaging: Proceedings of SPIE, San Jose, CA*, volume 4662, pages 57–69, 2002.

[8] Michael Dorr, Thomas Martinetz, Karl Gegenfurtner, and Erhardt Barth. Effects of gaze-contingent stimulation on eye movements with natural videos. *Perception Suppl.*, 33:134, 2004.

A user model of eye movements during visual search*

Wei Zhang
Computer Science Department
SUNY at Stony Brook
Stony Brook, NY 11794

Abstract

We present a computational framework modeling human eye movements in an object class detection task. The model combines state-of-the-art computer vision object class detection methods (SIFT features trained using AdaBoost) with a biologically plausible model of human eye movement to produce a sequence of simulated fixations, culminating with the acquisition of a target. We validated the model by comparing its behavior to the behavior of human observers performing the identical object class detection task (looking for a teddy bear among visually complex nontarget objects). We found considerable agreement between the model and human data in multiple eye movement measures, including number of fixations, cumulative probability of fixating the target, and scanpath distance.

This is joint work with Bing Yu, Hyejin Yang, Xin Chen, Dimitris Samaras and Gregory J. Zelinsky.

*Invited Talk

Inferring Relevance from Eye Movements: Feature Extraction

Jarkko Salojärvi[†], Kai Puolamäki[†], Jaana Simola[§], Lauri Kovanen[†]
Ilpo Kojo[§], Samuel Kaski[‡,†]

[†] Laboratory of Computer and Information Science
Neural Networks Research Centre
Helsinki University of Technology
P.O.Box 5400, FI-02015 HUT, Finland

[§] Center for Knowledge and Innovations Research
Helsinki School of Economics
Tammasaarenkatu 3, FI-00180 Helsinki, Finland

[‡] Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland

3 March 2005

Abstract

We organize a PASCAL EU Network of Excellence challenge for inferring relevance from eye movements, beginning 1 March 2005. The aim of this paper is to provide background material for the competitors: give references to related articles on eye movement modelling, describe the methods used for extracting the features used in the challenge, provide results of basic reference methods and to discuss open questions in the field.

1 Introduction

This technical report is written to complement the Inferring Relevance from Eye Movements challenge[1], one of the challenges partly funded by the EU network of excellence PASCAL. The challenge is organized in the form of a competition, where the contestants try to infer the relevance of a read document from the associated eye movement trajectory. We expect that the challenge will bring contributions to four different areas:

- Advances in machine learning methodology
- Establishing common practices for feature extraction in eye movements

[1]The Challenge has a web site at `http://www.cis.hut.fi/eyechallenge2005/`.

- Further the development of proactive user interfaces
- To learn of the psychology underlying eye movements in search tasks

The eye movement data is promising for advancing machine learning methods since it is very rich but noisy, and it is rather easy to collect in large quantities. The data is in the form of a time series which will pose challenges for optimal selection of features. For a simple case (Competition number 1), we will provide a comprehensive 22-dimensional set of eye movement features derived from the ones generally used in eye movement research (previously analysed in [31, 32]).

In psychological research of reading, it is common to segment the eye movement trajectory into fixations and saccades, and then compute summary measures of these modalities. The features used in Competition 1 are such summary measures. The controlled experimental setup used in the challenge makes it possible to test whether the established common practice is optimal for inferring relevance. In Competition 2 we give the full eye movement trajectory and the competitors can model it in any unorthodox way.

In information retrieval, relevance generally depends on the context, task, and individual competence and preferences of the user. Therefore relevance of articles suggested by a search engine could be improved by filtering them through an algorithm which models the interests of the user. This algorithm would be *proactive* [41]; it predicts the needs of the user and adapts its own behavior accordingly. Individual relevance can be learned from feedback given by the user. The usual way would be to ask after every document whether the user found it relevant, and to learn the user's preferences from the answers. However, giving this kind of explicit feedback is laborious, and people outside of research laboratories rarely bother. Alternatively, relevance can be inferred from implicit feedback derived traditionally from document reading time, or by monitoring other behavior of the user (such as saving, printing, or selecting of documents). The problem with the traditional sources is that the number of feedback events is relatively small. One of the motivations of the PASCAL challenge is to explore whether the traditional sources of implicit relevance information could be complemented with eye movements, and to find best methods for doing it.

In a typical information retrieval setup the user types in keywords to a search engine and is then given a list of titles of documents that possibly contain the information the user is looking for. Some of the documents suggested by the search engine will be totally irrelevant, some will handle the correct topic, and only few will be links to documents that the user actually will bother to read. Our experimental setting for collecting eye movement data was designed to simulate this natural situation, with the difference that in our case the relevance is known. By gathering data in a controlled setup we ensure that we know the ground truth, that is, the relevance associated with each eye movement trajectory. Machine learning methods can then be used for selecting a good set of features of eye movements, and for learning time series models to predict relevance of new measurements. If the eye movements contain any information about the relevance of a text, prediction should be possible. The modeling assumption behind our analysis is that attention patterns correlate with relevance; at the simplest, people tend to pay more attention to objects they find relevant or interesting.

2 Physiology of the Eye

Gaze direction is a good indicator of the focus of attention, since accurate viewing is possible only in the central *fovea* area (only 1–2 degrees of visual angle) where the density of photoreceptive cells is highly concentrated. For this reason, detailed inspection of a scene is carried out in a sequence of *saccades* (rapid eye movements) and *fixations* (the eye is fairly motionless). The trajectory is often referred to as a *scanpath*.

Information on the environment is mostly gathered during fixations, and the duration of a fixation is correlated with the complexity of the object under inspection. A simple physiological reason for this is that the amount of information the visual system is capable of processing is limited. During reading this complexity is associated with the frequency of occurrence of the words in general, and with how predictable the word is based on its context [29]. Naturally there are other factors affecting the reading pattern as well, such as different reading strategies and the mental state of the reader.

2.1 Eye movement details

Actually the eye does not lie completely still during fixations. In general we expect that the small movements during fixations will not play an important role in this challenge, since with the sampling rate of 50 Hz the average amount of samples from a fixation is around twelve. However, some basic knowledge on the fixations and saccades will be required if the competitors want to construct algorithms for fixation identification for Competition 2.

Clinical physiology text books [17] report that during fixation, the eye moves in an area which usually is less than 0.25 degrees of visual angle, meaning of the order of ten pixels in our experiment[2] (one should however also remember to take into account the measurement noise). During fixation, three different modes of movement can be separated: *tremor*, which is small amplitude (5–30 sec arc) and high frequency (30–100 Hz) oscillations, *drift*, which is slow velocity movement (1–8 min arc per second) and low frequency (<0.5 Hz), and *microsaccades*, low frequency (1–2 Hz) and small amplitude (1–8 min arc), saccade-like movements. Tremor and drift are commonly associated with the physiology of the eye, microsaccades on the other hand seem to have some cognitive basis [5, 19].

The saccades are ballistic, meaning that the target of the saccade will be decided before its initiation. The speed during a saccade depends on its length; for example during 5° saccade the peak velocity is around 260° per second, while during 20° saccade the peak velocity is around 660° per second. These characteristics are common to all people to the extent that one can use quantitative measurements of saccades to assess the function of the oculomotor system, to investigate the effects of drugs or lesions, and in some cases to aid diagnosis of disease or locating of lesions (see [10], for example).

The computation of a saccade requires some (latency) time in the fixation, meaning that fixations under 60 ms are not generally possible. However, it is possible to pre-program a sequence of saccades where the fixation duration will be shorter.

[2] with a subject distance of 60 cm from the 17" screen with a resolution of 1024x1280.

2.2 Pupillometry

In addition to eye movement features, the challenge also contains features computed from the pupil. There was some evidence in our experiments that the features correlated with relevance of the text [31]; the effect was very small at best, but it led us to discover the works reported in [16] or [2], where pupil diameter has been reported to increase as a sign of increased cognitive load.

The main function of pupil is to control the amount of light falling onto the retina. However, in addition to reflexive control of pupillary size there also seem to be tiny, cognitively related fluctuations of pupillary diameter ([2] reports interesting results that are discussed below). The so called *task-evoked pupillary response* (TERP) amplitudes appear to provide a good measure of the cognitive demands [2] for a wide variety of tasks (see Appendix for a brief note on TERPs).

Besides being a measure of cognitive demands of the task, the pupil width is also reported to vary due to different emotions. In [25], affective stimuli has been reported to cause systematical effects in subjects' physiological reactions and subjective experiences. The pupil size variation could therefore be used as implicit feedback signal for example in an affective computing interface [25].

3 Some literature

In this Section we give a brief introduction to literature on eye movements. The emphasis is on the areas which are relevant to the challenge: eye movements during reading and eye movements used as an implicit feedback channel.

3.1 Eye movements and reading

In a typical reading situation, the reader fixates on each word sequentially. Some of the words are skipped, some fixated twice and some trigger a *regression* to preceding words (approx. 15 % of the saccades). The reader is often not conscious of these regressions. The typical duration of fixations varies between 60–500 ms, being 250 ms on the average [21].

Research on eye movements during reading is a well-established field (see [29] for a good overview). In psychological literature, several models for reading have been proposed (most recent [6, 20, 30]). Models of eye movement control during reading differ mainly by the extent to which eye movements are assumed to be governed by lexical (high-level) processes over a simple default (low-level) control system assuming certain mean saccade lengths and fixation durations [39].

Currently the most popular model, so called E-Z Reader [30], concentrates on modeling reading at the basic level, as a series of sequential fixations occurring from left to right without regressions which are assumed to be associated with higher order cognitive processes. The durations of the fixations are correlated with word occurrence frequency, that is, the access time for the concepts concerning more rarely occurring words is longer than the access time for more frequently occurring words (however, similar correlations with word predictability and word length have also been reported). In a more recent publication [6] this correlation is extended to explain also regressions as occurring to those words which did not receive enough processing time during the first pass reading.

3.2 Eye movements and implicit feedback

Eye movements have earlier been utilized as alternative input devices for either pointing at icons or typing text in human-computer interfaces (see [15, 44]).

Use of eye movements as a source of implicit feedback is a relatively new concept. The first application where user interest was inferred from eye movements was an interactive story teller [38]. The story told by the application concentrated more on items that the user was gazing at on a display. Rudimentary relevance determination is needed also in [13], where a proactive translator is activated if the reader encounters a word which she has difficulties (these are inferred from eye movements) in understanding. A prototype attentive agent application (Simple User Interest Tracker, Suitor) is introduced in [22, 23]. The application monitors eye movements during browsing of web pages in order to determine whether the user is reading or just browsing. If reading is detected, the document is defined relevant, and more information on the topic is sought and displayed. Regretfully the performance of the application was not evaluated in the papers in any way. The (heuristic) rules for inferring whether the user is reading are presented in [4]. The eye movements have also been used as one feedback channel to identify critical driving events in intelligent driver assistance systems [24, 42].

The first analysis of eye movements in an information retrieval situation was published in [31, 32], where the experimental setup is quite similar to the Challenge. In [8] the goal was different: to investigate with quantitative measures how users behave in a real, less-controlled information retrieval task.

Implicit feedback information is also evaluated in usability studies[14, 7], where it is common to compute summary measures of eye movements on large areas of interest, such as images or captions of text (see [27] for an example study). The eye movements have also been used to give feedback of the subjective image quality [43].

4 Measurements

4.1 Experimental setup

Figure 1: An example of stimuli used in the experiments.

The structure of an assignment is as follows: a subject was first shown a

question (image 1 in Figure 1), and then a list of ten sentences (image 2 in Figure 1), one of which contained the correct answer (C). Five of the sentences were known to be irrelevant (I), and four relevant for the question (R). The subject was instructed to identify the correct answer and then press 'enter' (which ended the eye movement measurement) and then type in the associated number in the next screen (image 3 in Figure 1). The assignments were in Finnish, the mother tongue of the subjects.

The measurements were made for 11 subjects.

The *full training set* consists of 50 assignments, shown to all subjects. The lists were presented to the subjects in a randomized order. The measurements were carried out in sets of ten assignments, followed by a short break and re-calibration. Some of the assignments were excluded for technical reasons (e.g. the subject gave a wrong answer), resulting in less than 50 assignments per subject. In the challenge, the full training set is divided into a training and validation data set. The distribution of the correct answers in the full training data set is balanced, so that the correct answer appeared five times in the place of the first sentence, and so on.

Of the 11 subjects, seven were randomly chosen to take part in test data measurements. The *test set* consists of 180 assignments. To make cheating harder, all assignments within the test set are unique, and each assignment was shown to only one of the subjects. The locations of the relevant lines and correct answers in the test stimuli was randomly chosen, without balancing. The test data is constructed to be more real life-like, with less controlled questions and candidate sentences. It can therefore be expected that the classification rate is lower with the test data than with the training data.

4.2 Equipment

The device used for measuring eye movements was Tobii 1750 eye tracker[3], shown in Figure 2. The eye tracker is integrated into a 17" TFT monitor. The tracker illuminates the user with two near infrared diodes (they can be seen in Figure 2) to generate reflection patterns on the corneas of the user. A video camera then gathers these reflection patters as well as the stance of the user. Digital image processing is then carried out for extracting the pupils from the video signal. The systems tracks pupil location and pupil width at the rate of 50 Hz. The pupil locations can be mapped to gaze locations on the screen by calibrating the system; during the process the user needs to gaze at sixteen pre-defined locations on the screen.

The manufacturer reports the spatial resolution (frame-to-frame variation of the measured gaze point) to be 0.25 degrees and the average accuracy (bias error, deviation between the measured and actual gaze point of the user) of approximately 0.5 degrees. Additionally, the calibration deteriorates over time due to changes in the pupil size or if the eyes become dry. The associated drift of calibration is less than 0.5 degrees. The system allows free head motion in a cube of 30x15x20 cm at 60 cm from tracker. The resolution of the tracker is 1280x1024, and the recommended distance of the user from the display is 60 cm.

[3]Web pages at http://www.tobii.com. On 24 February 2005 a product description of the Tobii 1750 was available at http://www.tobii.com/downloads/Tobii_50series_PD_Aug04.pdf

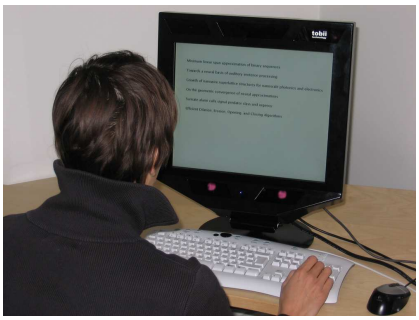


Figure 2: Eye movements of the subjects were measured with a Tobii 1750 eye tracker.

5 Feature Extraction

There are not many publications on the initial preprocessing of eye movement data (see [37] for an example). To our knowledge, the Tobii eyetracker does not preprocess the data[4].

5.1 Fixation Identification

Identifying fixations is still very much an open question within the eye tracking research, as there is no consensus of the method that best segments the eye movement trajectory (see [35] for discussion on the subject). Most of the eye movement measuring equipment manufacturers provide a *window-based* segmentation algorithm as a standard software. *Hidden Markov Model*-based algorithms have only recently gained some attention in the research area [33, 45].

5.1.1 Window-based Algorithms

In a window-based algorithm, a fixation is identified by drawing a square of x pixels around the currently measured gaze location. If the next measured gaze location falls within the block, it will be counted as a possible fixation. If in n consecutive gaze locations each falls within the block drawn around the gaze point preceding it, the n points will be counted as a fixation with a duration of n times the sampling interval (in our case 20 ms). In a Tobii eye tracker the standard setting is a 50-pixel window, with a time frame of 100 ms. For reading studies the manual recommends smaller window sizes. For the PASCAL challenge Competition 1, the fixations were computed using a 20 pixel window with a 80 ms time frame.

5.1.2 HMM-based Algorithms

The first application of Hidden Markov models (HMMs) to segment eye movement trajectories was [33], where a two-state HMM was applied. The model parameters were set manually, and the model was merely used for finding the most probable (Viterbi) path through the model for a given sequence in order to

[4]The Tobii however computes a *validity code* for each measurement, describing whether it tracks reliably both eyes or only one eye.

segment the trajectory. A more realistic application of the HMMs was presented in [45], where the parameters of a two-state HMM were learned from data.

Competitors taking part in the PASCAL Challenge Competition 2 may try to find the optimal segmentation method giving the best classification accuracy. Alternatively, they can of course decide to skip the segmentation part altogether.

5.2 Features for Competition 1

After segmenting the eye movement trajectory into fixations and saccades, they were assigned to the nearest word. After that, features for each word can be computed. All the features for the Competition 1 are listed in Table 1. We will next discuss the psychological justification behind the features.

The eye movement features used in psychological studies are often categorized into first-pass and second-pass measures, according to the order the region of text is encountered during reading. First-pass reading features are generally used as the primary measure of interest or as the measures of initial processing, whereas second-pass measures reflect the processes associated with re-analysis or "late processing" of the text region [29]. We expect the latter measures to play an important role in the challenge setup, for example in a case when the subject is choosing between two candidates of correct answers.

The eye movement features used in the challenge can additionally be divided into measures that are obtained from eye fixations, regressions, saccades, or pupil dilation data. In addition to the 22 features provided in the Competition 1, we will also briefly list some measures used in psychological studies for analysing the time series nature of the data, such as re-fixations and word skipping. These measures can be easily computed from the Competition 1 data.

Any single measure of processing would be an inadequate reflection of the reality of cognitive processing. To obtain a good description about the cognitive processes occurring during our task, a large number of different features need to be analysed. Features used in this paper and the challenge are listed in Table 1.

5.2.1 Fixation features

Typical measures of initial processing are first fixation duration (firstFixDur) and first-pass reading time or gaze duration (firstPassFixDur), which is the sum of all fixation durations on a region prior to moving to another region [3]. Additional measures for exploring early processes are the probability of fixating the target word (P1stFixation) when the region is initially encountered and the number of fixations received during first pass reading (FirstPassCnt). The duration of the fixation preceding the first fixation onto the current word (prevFixDur) and the duration of the next fixation after which the eyes moved to the next word (nextFixDur) were included in our analysis. In this paper, one measure of re-analysis or "late processing" was the probability that the word was fixated during second-pass reading (P2ndFixation). Measures covering all the fixations that landed on each word were also analysed. Mean fixation durations (meanFixDur), sums of all fixation durations on a word (totalFixDur) and the total number of fixations per word (fixCount) were computed, as well as the ratio between the total fixation duration and the total duration of fixations on the display (timePrctg).

5.2.2 Fixation position features

Landing position of first fixation on the word is used for exploring the early processing, whereas the launch site or the last location of the eyes before landing on the target word is used as a "control" for "parafoveal" preprocessing of the target word [3]. There is variability in where the eyes land on a word, but usually people tend to make their first fixation on a word about halfway between the beginning and the middle of a word [29]. This prototypical location is labelled as the *optimal viewing position*, where the word recognition time is minimized. Extensive research effort has been made to examine the consequences of making fixations at locations other than the optimal viewing position. It has been shown that the further the eyes land from the optimal position on a word the more likely there will be a refixation onto that word. We computed three measures that take the fixation position into account. The distance (in pixels) between the fixation preceding the first fixation on a word and the beginning of the word (prevFixPos), the distance of the first fixation on a word from the beginning of the word, and the launch site of the last fixation on the word from the beginning of the word (leavingPosition) were included.

5.2.3 Regressions

Approximately 10–15 % of fixations are regressions to previously read words. A common hypothesis is that eye movements during reading are mainly controlled by reasonably low-level processes in the brain, and higher level processes only interfere when something needs to be clarified. The second-pass measures such as regressions are therefore commonly accepted as indicators of higher-order cognitive processes. This may occur with a delay, since the transmission and processing of neural signals takes time.

In studies of reading it has been noted that the text difficulty has a strong influence on the number of regressions the readers make. Studies have also demonstrated that a regression was triggered when readers encountered a word indicating that their prior interpretation of a sentence was in error. Therefore it is likely that some of the regressions are due to comprehension failures [29].

Four regression measures were included in our set of features. We computed the number of regressions leaving from a word (nRegressionsFrom), the sum of durations of all regressions leaving from a word (regressDurFrom) and the sum of the fixation durations on a word during a regression (regressDurOn). It has been noted that sometimes the processing of a word "spills" on to reading the next word. Data analysis in [28] showed that most regressions originated from positions that were relatively close to a target word. In their dataset, of all the regressive saccades made within one line of text, 26 % came from within the same word (regressive refixations), 49.4 % came from the immediately following word, and 24.6 % came from more distant locations. We therefore included a binary feature (nextWordRegress) indicating whether the regression initiated from the following word.

5.2.4 Saccade features

Two saccade measures were included in the present paper. We computed the distance (in pixels) between the launch site of a saccade and its landing position,

when the fixation following the saccade was the first fixation onto a word (firstSaccLen) and when the fixation was the last fixation on a word (lastSaccLen).

5.2.5 Pupil features

There is evidence that the processing of complex sentences not only takes longer but it also produces a larger change in pupil diameter [2, 16]. Therefore two measures of pupil diameter were included in our analysis.

The mean horizontal pupil diameter during fixations on the current word was computed (pupilDiam1), as well as the maximum of pupil dilation within 0.5 – 1.5 seconds after encountering the word (pupilDiam2). The latter was the measure used in [16]. The measures were calibrated by subtracting the mean pupil diameter of the subject during the measurement.

5.2.6 Refixations

Refixation is a fixation to the currently processed word or text region. Some refixations occur because the gaze falls initially in a suboptimal place for processing the word, and a refixation takes the eyes to a more optimal viewing location [29]. The most frequent pattern is to first fixate near the beginning of the word followed by a fixation near the end of the word. Also contextual variables and incomplete lexical processing have been shown to have an effect on whether readers refixate on a current word. In [11] refixations were measured with sentences as the units of analysis. They computed the frequency and duration of reinspective fixations during the first reading of a sentence (reinspections). Hyönä [11] measured also the frequency and duration of looks back to a sentence that had already been read (look backs), and the frequency and duration of looks from a sentence back to an already read sentence (look froms). Reinspective and look-back fixations presented in [11] differ from regressions in that the saccadic direction is not decisive; rather, fixations that land on a previously fixated text region are defined either as reinspections (when reading parts of the currently fixated sentence) or look backs (when reading parts of a previously read sentence). All measures in [11] were computed as a ratio per character to provide adjustment for differences in length across sentences.

5.2.7 Skipping

There is experimental evidence that context has a strong effect on word skipping [29]. When the following words can be easily predicted from the context, they are more frequently skipped. Also high-frequency and short words are more easily skipped.

Note on the selected units of measures In psychology the most common unit of saccade lengths has been visual angle, which has the benefit of being independent of distance from stimuli. In studies of reading, saccade lengths have also been reported to scale with respect to font size. Both of these measures naturally demand that the subject's head is kept fixed throughout the measurements. Since the subject is allowed to move quite freely in our experiment (without losing too much accuracy), we will report saccade lengths in pixels, because converting them to angles or letter sizes would only add noise

to the measures due to movement of the subjects. The pixel measures with respect to each subject are comparable, since the stimuli were the same for all subjects, as was the the average distance of the subject to the display. Finally, the fixation identification algorithms provided by manufacturers of measuring equipment use the same units.

5.3 Features for Competition 2

In the challenge Competition 2, the raw eye movement data will be provided. The competitors are free to compute their own features from the x- and y-coordinates of gaze location and the pupil diameter. The given values are averages of the left and right eye.

6 Baseline Methods

6.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is one of the simplest means of classification, and it is discussed in most textbooks on applied statistics or multivariate techniques. The presentation here follows the one in [36].

The idea in LDA is to find new variables which are linear combinations of the original ones, such that different classes are discriminated as well as possible. Discrimination is measured by $SS_{between}/SS_{within}$, where $SS_{between}$ is the sum of squares between classes and SS_{within} the sum of squares inside a single class, defined by

$$SS_{within} = \sum_{g=1}^{G} \sum_{i=1}^{n_g} x_{gi}^2 \quad , \tag{1}$$

$$SS_{between} = \sum_{g=1}^{G} n_g (\bar{x}_g - \bar{x})^2 \quad , \tag{2}$$

where x_{gi} is the observation number i in class g, n_g is the number of observations in class $g = 1, \cdots, G$, $\bar{x}_g$ the mean of the observables in class g, and $\bar{x}$ the mean over all observations. In [36], the calculations needed to find optimal new axes are covered. We will next discuss how new observations are classified.

Let p_j be the prior probability and $f_j(x)$ the density function for class π_j. The observation x is allocated to the class π_j for which the probability of misclassification,

$$\sum_{i=1, i\neq j}^{G} p_i f_i(x) \quad , \tag{3}$$

is minimal. Clearly, this is the same as maximizing

$$\ln[p_j f_j(x)] \quad . \tag{4}$$

Assuming that x comes from a normal distribution, we get the classification rule (ignoring constants)

$$\text{argmax}_j \; [\ln p_j - 1/2 \ln |\Sigma_j| - 1/2 (x - \mu_j) \Sigma_j^{-1} (x - \mu_j)], \tag{5}$$

where Σ_j is the covariance matrix and μ_j the mean vector for class π_j in the training set.

6.2 Hidden Markov Models

In order to explain user behavior, the sequential nature of the reading process has to be modelled. Hidden Markov models are the most common methods for modeling sequential data. In eye movement research, hidden Markov models have earlier been used for segmenting the low-level eye movement signal to detect focus of attention (see [45]) and for implementing (fixed) models of cognitive processing [34], such as pilot attention patterns [9].

Hidden Markov models optimize the log-likelihood of the data Y given the model and its parameters Θ, that is, $\log p(Y|\Theta)$. The goal is to optimize the parameters of the model so that the distribution of the data is expressed as accurately as possible. HMMs are *generative models*; they attempt to describe the process of how the data is being generated. Therefore they can be said to *emit* (produce) observations.

Long-range time dependencies within the data are taken into account by adding hidden states to the model. The changes in the distributions of the emitted observations are associated with transitions between hidden states. The transitions (as well as the observation distributions) are modelled probabilistically. There exists a well-known algorithm for learning the HMMs, namely the Baum-Welch (BW) algorithm, if all the probabilities within the model are expressed using distributions which are within the exponential family [1]. Baum-Welch is a special case of Expectation-Maximization (EM) algorithm, and it can be proven to converge to a local optimum.

6.2.1 Simple Hidden Markov Model for Each Class

The simplest model that takes the sequential nature of data into account is a two-state HMM. We optimized one model individually for each class. In a prediction task the likelihood of each model is multiplied by the prior information on the proportions of the different classes in the data. As an output we get the maximum a posteriori prediction.

6.2.2 Discriminative Hidden Markov Models

In speech recognition, where HMMs have been extensively used for decades, the current state-of-the-art HMMs are discriminative. Discriminative models aim to predict the relevance $B = \{I, R, C\}$ of a sentence, given the observed eye movements Y. Formally, we optimize $\log p(B|Y, \Theta)$. In discriminative HMMs, a set of states or a certain sequence of states is associated with each class. This specific state sequence then gives the probability of the class, and the likelihood is maximized for the teaching data, versus all the other possible state sequences in the model [26]. The parameters of the discriminative HMM can be optimized with an extended Baum-Welch (EBW) algorithm, which is a modification of the original BW algorithm.

6.2.3 Discriminative Chain of Hidden Markov Models

A main difficulty in the information retrieval setup is that relevance is associated with titles, not with words in a title. For example, there are words in titles which are not needed in making the decision on whether the title is relevant or not. There could be many such non-relevant words in a sentence, and possibly only one word which is highly relevant. The situation thus resembles the setup in reinforcement learning: the reward (classification result) is only known in the end, and there are several ways to end in the correct classification.

In order to take into account the whole eye movement trajectory during a task, we model eye movements with a two-level discriminative HMM (see Figure 3). The first level models transitions between sentences, and the second level transitions between words within a sentence. Viterbi approximation is used to find the most likely path through the second level model (transitions between words in a sentence), and then the discriminative Extended Baum-Welch optimizes the full model (cf. [18, 40] for similar approaches).

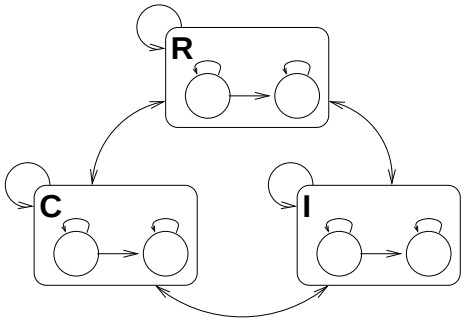


Figure 3: The topology of the discriminative chain of hidden Markov models.

In our implementation, the first level Markov model has three states, each state corresponding to one class of titles. Each of the three states in the first level have the following exponential family distributions:

1. A multinomial distribution emitting the relevance of the line, B. The parameters of this distribution were fixed, resulting in a discriminative Markov chain model in which each state corresponds to a known classification.

2. A *Viterbi distribution* emitting the probability of the sequence of words in a title.

The Viterbi distribution is defined by the probability of a Viterbi path trough a two-state Markov model forming the second level in our model. The two states of the second level model emit the observed word-specific distributions. The second level Viterbi distributions are further parameterized by the probabilities of beginning the sequence from that state (for example $\Pi^R = \pi_1^R, \pi_2^R$), and transition probabilities between states (e.g., a_{ij}^R, $i, j = 1, 2$). The second level Markov model is called a Viterbi distribution because when evaluating the emission probability only the most likely path over the two-state model is taken into

account (the Viterbi path). After fixing the path the resulting Viterbi distribution is (a fairly complex) exponential family distribution that can be trained with the EBW algorithm.

6.2.4 Voting

The Markov models produce probabilities for the relevance classes (I, R, C) for each viewed sentence. However, the users may look at a sentence several times, and the resulting probabilities need be combined in a process we call *voting*.

We constructed a log-linear model for combining the predictions. Assume that the sentence-specific probability distribution, $p(B|Y_{1...K})$, can be constructed from the probability distributions of the kth viewings of the sentence, $P(B|Y_k)$, (obtained as an output from a Markov model) as a weighted geometric average, $p(B|Y_{1...K}, \alpha) = Z^{-1} \prod_k p(B|Y_k)^{\alpha_{Bk}}$, where Z is a sentence-specific normalization factor and the parameters α_{Bk} are non-negative real numbers, found by optimizing the prediction for the training data. The predicted relevance of a sentence is then the largest of $p(I)$, $p(R)$, and $p(C)$.

It is also possible to derive a simple heuristic rule for classification by assuming that the decision of relevance is made only once while reading the sentence. We will call this rule maxClass, since for each sequence we will select the maximum of the predicted relevance classes. A simple baseline for the voting schemes is provided by classifying all the sequences separately (i.e., no voting).

7 Data analysis

Below we will carry out an example analysis of the challenge data. We apply Linear Discriminant Analysis to the eye movement data to obtain a first classification result, to get first visualizations of the data, and to select features that will be used in time series modeling, with HMMs and discriminative HMMs.

7.1 Linear Discriminant Analysis

Linear Discriminant Analysis is a simple linear method for analyzing data. Besides classification, the method can be used for visualization and feature set selection. It has not been developed for time series, however, and we apply it on feature vectors averaged over each sentence.

Averaged Features

Simple averaging of features presented in Table 1 would be against their spirit. The probabilities {3,4,18} are obtained by diving the sum of the features by the number of words in the sentence. Features {1,2,14,16,17,19,22} are commonly used as summary measures for larger areas of interest, and hence were merely added up. Features {5, 6, 7, 8, 9, 10, 11, 12, 13, 15} were computed as means, and for the pupil measures {20, 21} a maximum was taken (since in [16] the best effect was reported in the maximum of pupil dilation). Before analysing the data with LDA, the data was standardized.

Visualizing the Data with LDA

The data can be visualized by projecting them to the eigenvectors of the LDA (see Figure 4). The two eigenvectors define a hyperplane in the original feature space that best discriminates the classes. The visualization makes it possible to evaluate which classes will be harder to separate. Judging from the plot in Figure 4, it seems that relevant and irrelevant sentences will be hard to separate.

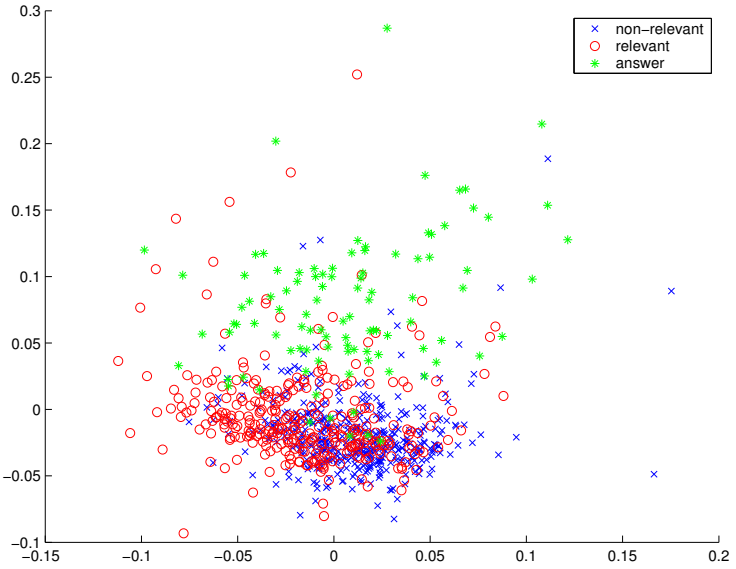


Figure 4: A visualization of the data using LDA.

Feature Set Selection with LDA

We may also plot the eigenvectors of the LDA in order to evaluate which components contribute most to the discriminating plane. Notice that if classification is not possible with LDA[5], the eigenvectors will be arbitrary. In our case, however, classification is possible as reported in Table 2. Judging from the eigenvectors plotted in Figure 5, it seems that less than ten features are sufficient.

7.2 Features for Time Series Analysis

Feature selection for the HMMs was carried out with the methods that use averaged data (LDA). In other words, we chose to model a representative set of features which can be used to construct the best discriminating averaged measures.

The resulting set of features were modeled with the following exponential family distributions: (1) One or many fixations within the word (binomial). (2) Logarithm of total fixation duration on the word (assumed Gaussian). (3)

[5]that is, the classification rate does not differ from a dumb classifier classifying all to the largest class

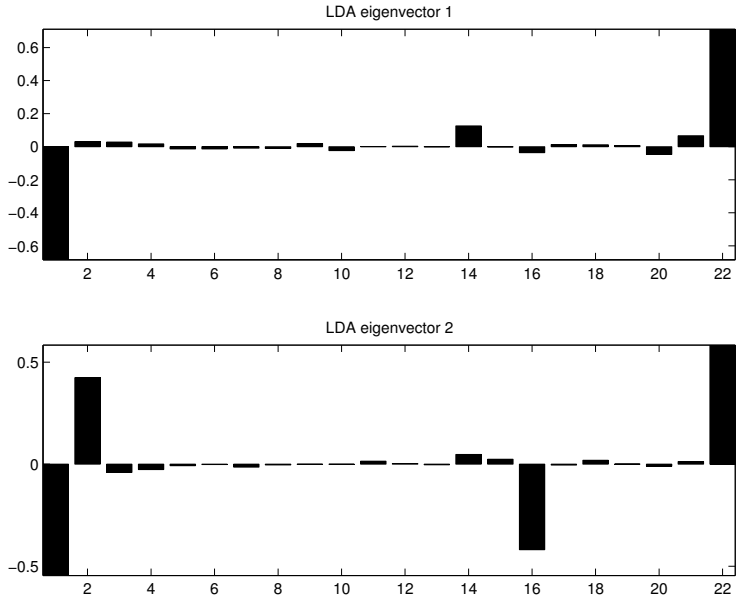


Figure 5: Eigenvectors of LDA. The histogram bars are loadings of the features, ordered according to Table 1.

Reading behavior (multinomial): skip next word, go back to already read words, read next word, jump to an unread line, or last fixation in an assignment.

7.3 Classification results

The prediction accuracy was assessed with 50-fold cross validation, in which each of the assignments was in turn used as a test data set. In order to test how the method would generalize to new subjects, we also ran an 11-fold cross validation where each of the subjects was in turn left out. The ultimate baseline is given by the "dumb model," which classifies all sentences to the largest class I. Table 2 lists the classification accuracies, that is, the fraction of the viewed sentences in the test data sets for which the prediction was correct. The methods generalize roughly equally well both to new assignments and to new subjects. The performance of the two different voting methods (log-linear and maxClass) seems to be nearly equal, with log-linear voting having a slight advantage.

Table 3 shows the confusion matrix of the discriminative HMMs. Correct answers (C) are separated rather efficiently. Most errors result from misclassifying relevant sentences (R) as irrelevant (I). It is also possible to compute precision and recall measures common in IR, if the correct answers are treated as the relevant documents. The resulting precision rate is 90.1 % and recall rate 92.2 %.

8 Discussion

The physiology and psychology of eye movements has been studied quite extensively. However, the improved multimodal interfaces, combined with proactive

information retrieval, provide us with a whole new setup. The eye movements are a rich, complex, and potentially very useful time series signal. Efficient extraction of relevance information from it is not trivial, however, and requires development and application of advanced machine learning methods.

The features used in eye movement research have been based mostly on the segmentation of the eye movement trajectory to fixations and saccades. This segmentation, though useful, is neither unique nor always optimal. The optimal set of features is likely to depend on the task at hand. One of the goals of Competition 2 of this Challenge is to find and propose a new set eye movement features, not necessarily based on the division to fixations and saccades, for use in eye movement studies and proactive applications.

In the study of eye movements in psychology the basic goal is to understand the underlying psychological processes. Our objective is different and more application-oriented: we want to extract maximal amount of useful information from the real-world eye movement signal, to be used in proactive information retrieval. Our approach also differs from usability studies, another common application of eye movement analysis, where the objective has been to analyze qualitatively and quantitatively the behavior of a user when she for instance visits a web site. The quantitative measures have been mostly based on fixation durations and eye scan patterns. This Challenge differs from much of the prior work in its application and experimental setup (information retrieval task where the ground truth is known) and in the use of advanced probabilistic methods optimized for the task at hand (relevance extraction).

The Challenge will hopefully result in a toolbox of new machine learning methods and a set of features, optimal for extracting relevance information from the real world eye movement signals.

We are looking forward to an interesting competition and wish all participants the best of success!

Acknowledgments

This work was supported by the Academy of Finland, decision no. 79017, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. The authors would like to thank people in the PRIMA project for useful discussions, and acknowledge that access rights to the data sets and other materials produced in the PRIMA project are restricted due to other commitments.

The authors are greatly indebted to Maria Sääksjärvi and Johanna Gummerus, Center for Relationship Marketing and Service Management at Swedish School of Economics and Business Administration at Helsinki, for providing us the measurement time. We would also like to thank Kari-Jouko Räihä, Aulikki Hyrskykari and Päivi Majaranta from Tampere Unit for Computer-Human Interaction for useful discussions and measurement time.

References

[1] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic func-

tions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, February 1970.

[2] Jackson Beatty and Brennis Lucero-Wagoner. The pupillary system. In John T. Cacioppo, Louis G. Tassinary, and Gary G. Berntson, editors, *Handbook of Psychophysiology*, chapter 6. Cambridge University Press, Cambridge, UK, 2000.

[3] Manuel G. Calvo and Enrique Meseguer. Eye movements and processing stages in reading: Relative contribution of visual, lexical and contextual factors. *The Spanish Journal of Psychology*, 5(1):66–77, 2002.

[4] Christopher Campbell and Paul Maglio. A robust algorithm for reading detection. In *Workshop on Perceptive User Interfaces (PUI '01)*. ACM Digital Library, November 2001. ISBN 1-58113-448-7.

[5] Ralf Engbert and Reinhold Kliegl. Microsaccades uncover the orientation of covert attention. *Vision Research*, 43:1035–1045, 2003.

[6] Ralf Engbert, André Longtin, and Reinhold Kliegl. A dynamical model of saccade generation in reading based on spatially distributed lexical processing. *Vision Research*, 42:621–636, 2002.

[7] Joseph H. Goldberg, Mark J. Stimson, Marion Lewenstein, Neil Scott, and Anna M. Wichansky. Eye tracking in web search tasks: design implications. In *ETRA '02: Proceedings of the symposium on Eye tracking research & applications*, pages 51–58. ACM Press, 2002.

[8] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in WWW search. In *Proceedings of SIGIR'04*, pages 478–479. ACM Press, 2004.

[9] Miwa Hayashi. Hidden Markov models to identify pilot instrument scanning and attention patterns. In *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, pages 2889–2896, 2003.

[10] S.B. Hutton, I. Cuthbert, T.J. Crawford, C. Kennard, T.R.E. Barnes, and E.M. Joyce. Saccadic hypometria in drug-naïve and drug-treated schizophrenic patients: A working memory deficit? *Psychophysiology*, 38:125–132, 2001.

[11] J. Hyönä, R.F.(Jr) Lorch, and J.K. Kaakinen. Individual differences in reading to summarize expository text: Evidence from eye fixation patterns. *Journal of Educational Psychology*, 94:44–55, 2002.

[12] J. Hyönä, J. Tommola, and A.M. Alaja. Pupil dilation as a measure of processing load in simultaneous interpreting and other language tasks. *Quarterly Journal of Experimental Psychology*, 48A:598–612, 1995.

[13] Aulikki Hyrskykari, Päivi Majaranta, and Kari-Jouko Räihä. Proactive response to eye movements. In G. W. M. Rauterberg, M. Menozzi, and J. Wesson, editors, *INTERACT'03*. IOS press, 2003.

[14] R.J.K. Jacob and K.S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises (section commentary). In J. Hyona, R. Radach, and H. Deubel, editors, *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, pages 573–605. Elsevier Science, Amsterdam, 2003.

[15] Robert J. K. Jacob. *Eye tracking in advanced interface design*, pages 258–288. Oxford University Press, 1995.

[16] Marcel Adam Just and Patricia A. Carpenter. The intensity dimension of thought: Pupillometric indices of sentence processing. *Canadian Journal of Experimental Psychology*, 47(2):310–339, 1993.

[17] K.J. Ciuffreda KJ and B. Tannen. *Eye Movement Basics for the Clinician*. Mosby Yearbook, St. Louis, 1995.

[18] Mikko Kurimo. *Using Self-Organizing Maps and Learning Vector Quantization for Mixture Density Hidden Markov Models*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 1997.

[19] Jochen Laubrock, Ralf Engbert, and Reinhold Kliegl. Microsaccade dynamics during covert attention. *Vision Research*, 45:721–730, 2003.

[20] Gordon E. Legge, Timothy S. Klitz, and Bosco S. Tjan. Mr. chips: An ideal-observer model of reading. *Psychological Review*, 104(3):524–553, 1997.

[21] Simon P. Liversedge and John M. Findlay. Saccadic eye movements and cognition. *Trends in Cognitive Sciences*, 4(1):6–14, 2000.

[22] Paul P. Maglio, Rob Barrett, Christopher S. Campbell, and Ted Selker. Suitor: an attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 169–176. ACM Press, 2000.

[23] Paul P. Maglio and Christopher S. Campbell. Attentive agents. *Commun. ACM*, 46(3):47–51, 2003.

[24] Bradford Miller, Chung Hee Hwang, Kari Torkkola, and Noel Masseya. An architecture for an intelligent driver assistance system. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 639–644, June 2003.

[25] Timo Partala, Maria Jokiniemi, and Veikko Surakka. Pupillary responses to emotionally provocative stimuli. In *Proceedings of Eye Tracking Research and Applications (ETRA2000)*, pages 123–129. ACM press, 2000.

[26] D. Povey, P.C. Woodland, and M.J.F. Gales. Discriminative MAP for acoustic model adaptation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03)*, volume 1, pages 312–315, 2003.

[27] `http://www.poynter.org/eyetrack2000/`.

[28] R. Radach and G.W. McConkie. Determinants of fixation positions in words during reading. In G. Underwood, editor, *Eye Guidance in Reading and Scene Perception*, pages 77–100. Elsevier, Oxford, 1998.

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

[30] Erik D. Reichle, Alexander Pollatsek, Donald L. Fisher, and Keith Rayner. Toward a model of eye movement control in reading. *Psychological Review*, 105(1):125–157, 1998.

[31] Jarkko Salojärvi, Ilpo Kojo, Jaana Simola, and Samuel Kaski. Can relevance be inferred from eye movements in information retrieval? In *Proceedings of WSOM'03, Workshop on Self-Organizing Maps*, pages 261–266. Kyushu Institute of Technology, Kitakyushu, Japan, 2003.

[32] Jarkko Salojärvi, Kai Puolamäki, and Samuel Kaski. Relevance feedback from eye movements for proactive information retrieval. In Janne Heikkilä, Matti Pietikäinen, and Olli Silvén, editors, *workshop on Processing Sensory Information for Proactive Systems (PSIPS 2004)*, pages 37–42, Oulu, Finland, 14-15 June 2004.

[33] Dario D. Salvucci and John R. Anderson. Tracing eye movement protocols with cognitive process models. In *Proceedings of the Twentieth Annual Conference of the Cognitive Society*, pages 923–928, Hillsdale, NJ, 1998. Lawrence Erlbaum Associates.

[34] Dario D. Salvucci and John R. Anderson. Automated eye-movement protocol analysis. *Human-Computer Interaction*, 16:39–86, 2001.

[35] D.D. Salvucci and J.H. Goldberg. Identifying ixations and saccades in eye-tracking protocols. In *Proceedings of the Eye Tracking Research and Applications Symposium 2000 (ETRA2000)*, 2000.

[36] Subhash Sharma. *Applied Multivariate Techniques*. John Wiley & Sons, Inc., 1996.

[37] Dave M. Stampe. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments & Computers*, 25(2):137–142, 1993.

[38] India Starker and Richard A. Bolt. A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 3–10. ACM Press, 1990.

[39] M.S. Starr and K. Rayner. Eye movements during reading: some current controversies. *Trends in Cognitive Sciences*, 5(4):156–163, 2001.

[40] A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 11–18, San Francisco, CA, 1993. Morgan Kaufmann.

[41] David Tennenhouse. Proactive computing. *Commun. ACM*, 43(5):43–50, 2000.

[42] Kari Torkkola, Noel Masseya, Bob Leivian, Chip Wood, John Summers, and Snehal Kundalkar. An architecture for an intelligent driver assistance system. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 81–85, June 2003.

[43] Tero Vuori, Maria Olkkonen, Monika Pölönen, Ari Siren, and Jukka Häkkinen. Can eye movements be quantitatively applied to image quality studies? In *Proceedings of the third Nordic conference on Human-computer interaction*, pages 335–338, 2004.

[44] David J. Ward and David J.C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418:838, 2002.

[45] Chen Yu and Dana H. Ballard. A multimodal learning interface for grounding spoken language in sensory perceptions. In *Proc. ICMI'03*. ACM, 2003. To appear.

A Notes on TERP

Because TERP amplitudes appear to be independent of baseline pupillary diameter, it is possible to compare the amplitude of TERPs obtained in different laboratories. Analysis of pupillometric data in memory storage and recall tasks have shown that there is variation in peak pupillary dilation as a function of the length of the target string to be stored or recalled. The item difficulty in memory tasks has also been associated with greater pupillary dilations.

There is evidence of response and movement-related pupillary responses. Results from experiments where immediate or delayed response selection and preparation were studied indicated that the rate of pupil dilation was inversely proportional to the length of the foreperiod preceding the imperative stimulus. It was shown that the pupil dilations were greater in Go-trials than dilations to No-Go stimuli in both immediate- and delayed-response conditions. Additionally, both peak pupil diameter and peak latency have been found to vary with the complexity of movements in motor tasks.

It has been reported that pupil dilations are elicited not only by external stimuli but also by a stimulus mismatch or by an orientation to a task important stimuli. An inverse relationship has been found between pupil amplitude and probability. Pupil dilations were found to be larger in amplitude and longer in latency for stimuli with low probability of occurrence.

TERP amplitude is also a sensitive and reliable reporter of differences in the structure of cortical language processing and decision. In a letter matching task, physically identical letter pairs evoked smaller TERPs than did pairs identical only at the level of naming. [16] found that more complex sentence types produced larger changes in pupil diameter. [12] reported, that increases in semantic demands of sentence processing resulted in increases of the TERP.

Table 1: Features

	Feature	Description
1	**fixCount**	Total number of fixations to the word
2	**FirstPassCnt**	Number of fixations to the word when the word is first encountered
3	**P1stFixation**	Did a fixation to a word occur when the sentence that the word was in was read for the first time ('1' or '0')
4	**P2ndFixation**	Did a fixation to a word occur when the sentence that the word was in was read for the second time ('1' or '0')
5	**prevFixDur**	Duration of the previous fixation when the word is first encountered
6	**firstFixDur**	Duration of the first fixation when the word is first encountered
7	**firstPassFixDur**	Sum of durations of fixations to a word when it is first encountered
8	**nextFixDur**	Duration of the next fixation when the gaze initially moves on from the word
9	**firstSaccLen**	Distance (in pixels) between the launching position of the previous fixation and the landing position of the first fixation
10	**lastSaccLen**	Distance (in pixels) between the launching position of the last fixation on the word and the landing point of the next fixation
11	**prevFixPos**	Distance (in pixels) between the fixation preceding the first fixation on a word and the beginning of the word
12	**landingPosition**	Distance (in pixels) of the first fixation on the word from the beginning of the word
13	**leavingPosition**	Distance (in pixels) between the last fixation before leaving the word and the beginning of the word
14	**totalFixDur**	Sum of all durations of fixations to the word
15	**meanFixDur**	Mean duration of the fixations to the word
16	**nRegressionsFrom**	Number of regressions leaving from the word
17	**regressDurFrom**	Sum of durations of fixations during regressions initiating from the word
18	**nextWordRegress**	Did a regression initiate from the following word ('1' or '0')
19	**regressDurOn**	Sum of the durations of the fixations on the word during a regression
20	**pupilDiam1**	Mean of pupil diameter during fixations on the word (minus mean pupil diameter of the subject during the measurement)
21	**pupilDiam2**	Maximum of pupil dilation within 0.5 – 1.5 seconds after encountering the word (minus mean pupil diameter of the subject during the measurement)
22	**timePrctg**	Total fixation duration on a word divided by the total duration of fixations on the display

Table 2: Performance of the different models in predicting relevance of the sentences. Differences between LDA and dumb classifier, and HMM and LDA tested significant (McNemar's test), as well as difference between discriminative HMM and simple HMMs (with leave-one-assignment-out cross validation) Left column: obtained by 50-fold cross-validation where each of the assignments was left out in turn as test data. Right column: Obtained by 11-fold cross-validation where each of the subjects was left out in turn to be used as test data.

Method	**Accuracy (%)** (leave-one-assignment-out)	**Accuracy (%)** (leave-one-subject-out)
Dumb	47.8	47.8
LDA	59.8	57.9
simple HMMs(no vote)	55.6	55.7
simple HMMs(maxClass)	**63.5**	**63.3**
simple HMMs(loglin)	**64.0**	**63.4**
discriminative HMM(loglin)	**65.8**	**64.1**

Table 3: Confusion matrix showing the number of sentences classified by the discriminative HMM, using loglinear voting, into the three classes (columns) versus their true relevance (rows). Cross-validation was carried out over assignments. The percentages (in parentheses) denote row- and column-wise classification accuracies.

	Prediction		
	I (62.4 %)	R (61.8 %)	C (90.1 %)
I (77.3 %)	1432	395	25
R (43.6 %)	845	672	24
C (92.2 %)	17	21	447