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

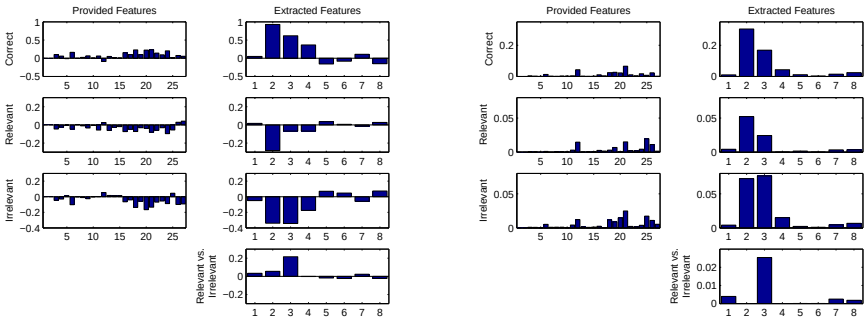
2.2 Statistical Analysis

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.