Quality Enhancement in Written Examinations by Automatic Recognition of Correction Results

Arian Mehmanesh, Stephan Günther*, Johannes Naab*, Maurice Leclaire*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: arian.mehmanesh@tum.de, guenther@tum.de, naab@net.in.tum.de, leclaire@in.tum.de

***Abstract*—An examination score determined by human correctors can be erroneous in multiple ways. In this case the focus is on errors caused by miscalculating the score manually. The goal is to estimate a rate of exams with calculation mistakes and determine which factors are likely to increase this error rate when designing an exam. To achieve this, 525 sample exams are digitized by hand with help of an automated system detailed herein. Afterwards, the digital exam scores are scanned for errors automatically. Statistical analysis of these errors yields the following results:**

The quote of exam points miscalculated in this sample is determined to be 4.2%, resulting in a confidence interval between between 2.7% and 6.4% (95% conf. level). On average, the correctors made an error when a problem has 10.4 subproblems distributed over 3.9 pages. The mean impact of an error is 0.8% of the total credits. Considering the time it takes to manually add the points and determine the grade, in combination with this error rate, automated score calculation systems for exams are proposed as a solution.

***Index Terms*—examinations, human error, correction, miscalculation, scoring**

1. Introduction

Human error is unavoidable in the correction of examinations by hand. While these faults can occur by overlooking correct answers or grading similar responses differently, the last step of every evaluation is to sum all points and determine the final grade. This paper focuses on quantifying the errors made in calculating the exam scores since they are measurable and comparable across all forms and topics of examinations. These arithmetical errors are also chosen because they are computationally detectable.

To achieve this, a sample examination is analyzed in its entirety for falsely calculated credits. This analysis requires an optimized interface to facilitate quick acquisition of the recognized scores. After the recognition of handwritten exam scores the correctors' faults are detected by an automated validation algorithm.

After all errors are found, the percentage of miscalculated exams and by how much the error deviates from the correct credits on average are of interest. Additionally, relations between the error rate of a problem and the count of its subproblems (or how many pages the problem spans, respectively) are inspected.

The content of this paper is structured as follows: In Section 2 a related study is presented. The dataset used in this project is detailed in Section 3.1, followed by a description of the software used to extract numerical score values from exam images in Section 3.2. These values are statistically analyzed in Section 3.3. The results of this analysis are presented in Section 4, amounting to the conclusion in Section 5.

2. Related work

Studies discussing errors in examinations are common, but for comparison to this analysis they are required to differentiate between score calculation mistakes and other errors. Phillips & Weathers have analyzed 5017 standardized tests (Stanford Achievement Test) in 1958 [1]. They quantified and distinguished different types of errors, like correctors not following the instructions or falsely computing the final grade based on the students total score. The focus of this paper, the incorrect summing up of scores, was observed aswell but referred to as "counting error". Out of the total 5017 tests, 630 of them were miscounted (13 %). This was the most prominent fault, causing 45 % of all errors.

3. Methods

After the description of the dataset used for this project, the two main parts of the methodology are detailed. They consist of the interface used for recognizing the written scores and how these determined values are analyzed for errors.

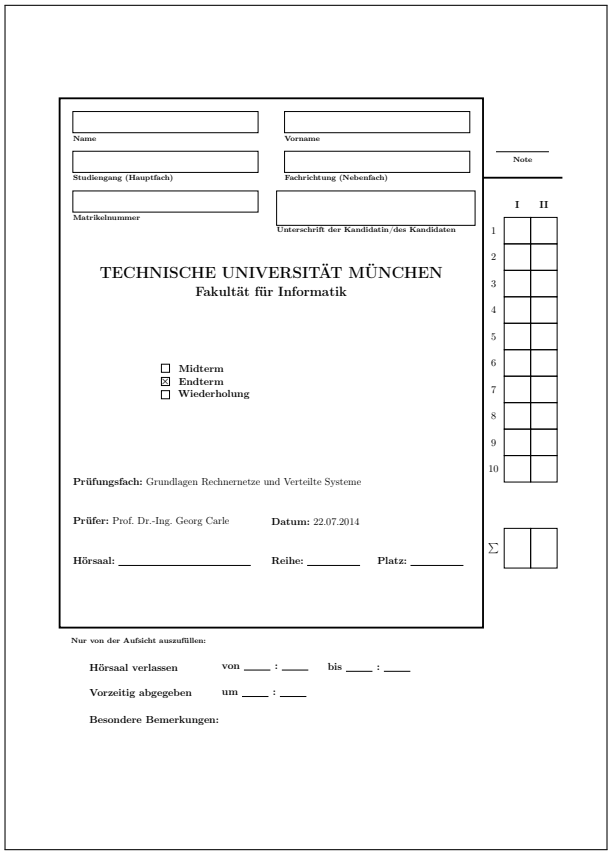
3.1. Dataset

The analyses herein are based on a digitized endterm examination of 2014 provided as scanned images. It was held at the Technical University of Munich (TUM) on the topic of "Basics in Networking and Distributed Systems" [2], consisting of 525 individual exams.

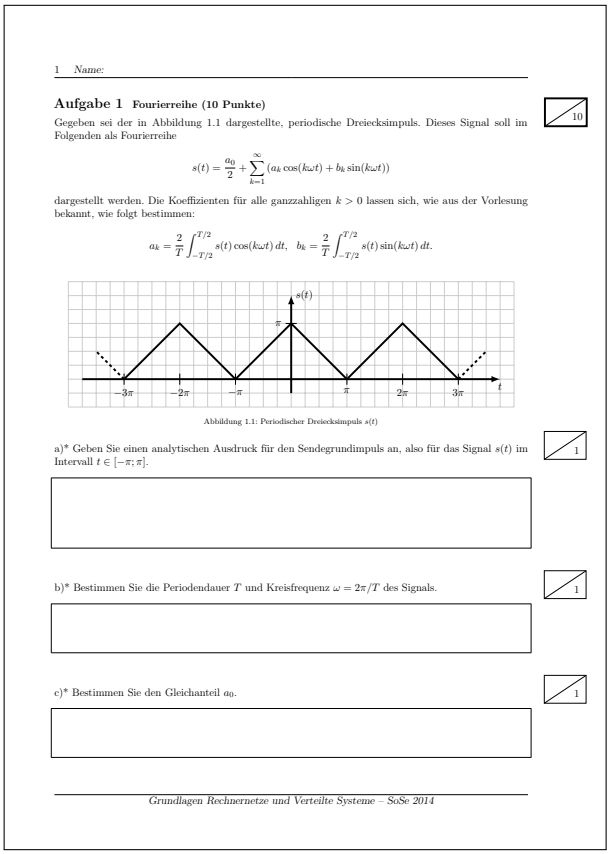
Figure 1a shows the front page of the exam, the points noted here sum up to yield the final grade. The first problem of the exam is demonstrated in Figure 1b, where the scores of the subproblems are added and are written in the top box. This result of problem 1 is carried over to the front page (Fig. 1a).

Every exam consists of 68 score boxes, resulting in 35700 total boxes available. This examination was evaluated twice by the correctors, in a first and second run. The result of the second run determines the final grade.

doi: 10.2313/NET-2019-10-1_10



(a) Front page score boxes and total sum.



(b) Problem score box with subproblems.

Figure 1: Sample pages from the studied exam.

3.2. Recognition

The optical character recognition (OCR) of the written credits is performed manually. Automated OCR or interpretation by a machine learning approach exceeds the scope of this analysis. To optimize this process it is necessary to automate the displaying of score boxes to the reviewer and recording the recognized score for each problem. Additionally, metadata should be tracked for every box, such as the page on which the box was located and the time interval it took the reviewer to recognize and enter its numerical values. For the implementation of this automation the programming language Python [3] is used due to its ease of use and legibility.

The structure of the program can be reduced to the Model-View-Controller pattern [4] which allows these three components to be detailed separately.

3.2.1. Model. The model replicates the dataset and consists of the whole `ExamBatch`, a single `Exam` and the individual `Problem` which represents a score box. An `ExamBatch` manages a list of `Exams`, whereas an `Exam` stores a tree of `Problems`.

In the example of Figure 2, the root node of the tree is the score box of the total exam credits. Subordinated are `Problem` 1 and 2 on the front page, which are the scores of `Problem` 1 and 2 carried over from the inside of the paper exam sheet. Each `Problem` contains two subproblems.

The tree structure is chosen because every `Problem` can have an arbitrary amount of subproblems in a generic paper exam.

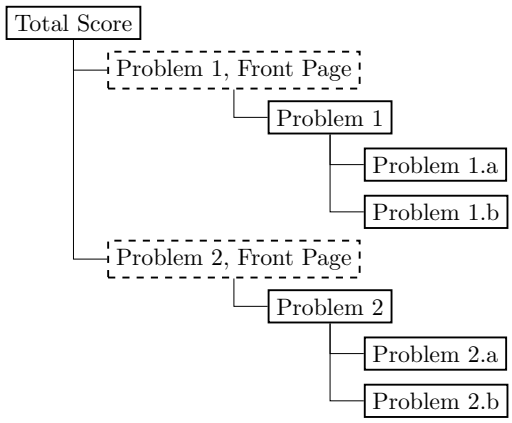


Figure 2: An example tree of Problems.

Dashed lines indicate a score being carried over, not computed.

To iterate through the problems, a depth-first search approach is used [5] as it is similar to the way a paper exam is usually corrected.

3.2.2. View. The graphical interface for the user is kept simple to support fast recognition.

As shown in Figure 3 of the program in execution, a cropped score box and the user input can be seen. The credits of the first correction pass are marked in red, the

doi: 10.2313/NET-2019-10-1_10

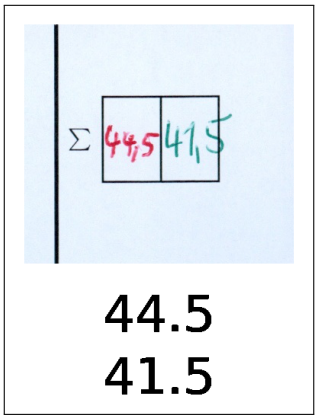


Figure 3: View of the User Interface.

second pass in green. At this stage, the user has entered the values of both scores below the box. After this step the software jumps to the next box immediately.

3.2.3. Controller. The responsibility of the controller is to manage the control flow. The following pseudocode is used to describe its algorithm.

```
load all exam scans into an ExamBatch
for every Exam in the ExamBatch do
  repeat
    display the next Problem (depth-first search)
    start the timer
    await user input
    stop the timer
    store user input and metadata
  until no Problem left in Exam
  store Exam as JSON file [6]
end for
```

Figure 4: The control flow in pseudocode.

As indicated in Figure 4, the controller basically performs a slideshow of score boxes awaiting user input of float values at every step.

3.3. Analysis

To detect miscalculations in the JSON files stored in the recognition phase (Section 3.2), a second python program is used. It is tasked with iterating through the digitized exam data and recompute the scores for every exam. If a mismatch between the calculated and the written credits is detected, an error is recorded. Since this task is significantly less complex than the first program, it is not detailed further.

The main target estimation to be provided by this document is the likelihood p of an exam being falsely graded. Every exam can assume two states, namely being correctly or incorrectly scored. This leads to the assumption of a binomial distribution with parameter p. There are multiple methods for estimating a confidence interval (CI) for a binomial distribution. While the Wald interval method is very prevalent in textbooks, Vollset [7] discourages its use and recommends the Wilson score interval with continuity correction. This method can be applied, because the binomial distribution can be approximated by a normal distribution for large sample sizes.

Let $\hat{p} = \frac{22}{525}$ be the realisation of p in this sample, $n = 525$:

$$\begin{aligned} n\hat{p}(1-\hat{p}) &\geq 9 \\ &\approx 21 \geq 9 \end{aligned} \tag{1}$$

As shown in the Equation (1) our sample is large enough for this continuity correction [8]. The Wilson score interval with continuity correction is determined by [9]:

$$\begin{aligned} L &= \frac{2n\hat{p} + z^2 - 1 - z\sqrt{z^2 - 2 - 1/n + 4\hat{p}(n(1-\hat{p}) + 1)}}{2(n + z^2)} \\ U &= \frac{2n\hat{p} + z^2 + 1 + z\sqrt{z^2 + 2 - 1/n + 4\hat{p}(n(1-\hat{p}) - 1)}}{2(n + z^2)} \end{aligned} \tag{2}$$

Where L is the lower and U the upper bound of the confidence interval. For a confidence level of 95% the value z is the $1 - \frac{1-0.95}{2}$ quantile of the standard normal distribution (Φ is its cumulative distribution function):

$$\begin{aligned} z &= \Phi^{-1}(1 - \frac{1 - 0.95}{2}) \\ &= 1.96 \end{aligned}$$

Equation (2) is later used for the computation of the CI.

In an attempt to interpret the nature of the mistakes made by the correctors, the errors are further dissected. The following attributes of an error are averaged:

1) amount of subproblems that had to be added
2) number of pages the mistake was distributed over
3) absolute offset of the noted score versus the correct one

Finally, the average time needed to recognize a score box or a whole exam is determined. This assesses the effort of a human reading score boxes.

4. Results

The confidence interval of the likelihood of an exam being wrongly corrected is between 2.7% and 6.4% with a mean estimate of 4.2%.

On average, an error is based on 10.5 subproblems that were erroneously added. Furthermore, these values were generally added over 3.9 pages (requiring avg. 1.6 physical page turns). Errors deviate from the correct score by 0.7 points (0.8% of the total score).

All 22 detected errors result from falsely summing subproblem points to a problem, none were made adding the credits on the front page. The score of a Problem was never incorrectly carried over to the front page.

Recognizing a single score box takes about 1.6 seconds, resulting in 85 seconds total per exam. Since a program to facilitate recording of credits is used, these time measures do not include:

1) flipping through the pages
2) localizing score boxes
3) computing the addition

doi: 10.2313/NET-2019-10-1_10

4) fixing own mistakes in this process

Thus the measured time of over 12 hours total is significantly lower than the time required by a human correcting paper exams by hand.

5. Conclusion and future work

Concerning the rate of falsely added exam scores, an interval of 2.7% to 6.4% is high (95% CI). Assuming this value is representative for all university exams and a student participates in four exams on average per semester, from 48% to 80% of bachelor students have at least one of their exam credits miscalculated. Although the impact of 0.8% of the score in these errors seems to be low, exam grading is discrete. This leads to such a deviation having either no effect or result in a significant grade change.

The results of this study indicate that exams containing problems with many subproblems or problems which are distributed over several pages are more prone to error. Further research is needed to validate this claim.

Recognizing all credit values of an exam took 85 seconds, so the total time required for evaluating all exams amounts to over 12 hours. As explained in Section 4, correcting a paper exam without the tools for automation described herein takes significantly longer.

There is a solution for minimizing computational errors and drastically decreasing the required time to evaluate exams. Phillips & Weathers have already pointed out in 1958 that "An alternative would be to have all standardized tests machine-scored" [1]. Automating the recognition and addition of exam scores in the present and future is inevitable and extended research to enhance such software is recommended.

References

[1] B. N. Phillips and G. Weathers, "Analysis of errors made in scoring standardized tests," *Educational and Psychological Measurement*, vol. 18, no. 3, 1958.

[2] G. Carle, "Vorlesung Grundlagen Rechnernetze und Verteilte Systeme," https://www.net.in.tum.de/teaching/ss14/vorlesungen/vorlesung-rechnernetze-und-verteilte-systeme/index.html/, 2014, [Online; accessed 12-June-2019].

[3] Python Software Foundation, "Python language reference, version 3.7," https://docs.python.org/3.7/, 2019, [Online; accessed 17-June-2019].

[4] G. E. Krasner and S. T. Pope, "A cookbook for using the model-view controller user interface paradigm in smalltalk-80," *J. Object Oriented Program.*, vol. 1, no. 3, pp. 26–49, Aug. 1988.

[5] K. Mehlhorn and P. Sanders, *Algorithms and Data Structures: The Basic Toolbox*. Springer, Oct. 2007. [Online]. Available: https://people.mpi-inf.mpg.de/~mehlhorn/ftp/Mehlhorn-Sanders-Toolbox.pdf

[6] ECMA International, "The json data interchange syntax," http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf, Dec. 2017, [Online; accessed 18-June-2019].

[7] S. E. Vollset, "Confidence intervals for a binomial proportion," *Statistics in Medicine*, vol. 12, no. 9, pp. 809–824, 1993.

[8] M. Sachs, *Wahrscheinlichkeitsrechnung und Statistik*. Hanser Fachbuchverlag, Sep. 2003.

[9] R. G. Newcombe, "Two-sided confidence intervals for the single proportion: comparison of seven methods," *Statistics in Medicine*, vol. 17, no. 8, pp. 857–872, 1998.