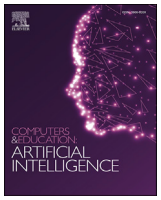
Computers and Education: Artificial Intelligence 5 (2023) 100168

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence



Automatic proficiency scoring for early-stage writing

Michael Riis Andersen [a,*], Kristine Kabel [b], Jesper Bremholm [c], Jeppe Bundsgaard [b], Lars Kai Hansen [a]

[a] DTU Compute, Technical University of Denmark, Richard Petersens Plads, Kongens Lyngby 2800, Denmark
[b] School of Education, Aarhus University, Tuborgvej 164, København NV, 2400, Denmark
[c] National Centre for Reading, Danish University Colleges, Humletorvet 3, København V, 1799, Denmark

ARTICLE INFO

Keywords:
Early-stage literacy
Machine learning
Natural language processing
Rasch models
Writing proficiency
Automatic scoring
Danish
Low-resource languages

ABSTRACT

In this work, we study the feasibility of using machine learning and natural language processing methods for assessing writing proficiency in Danish with respect to text construction, sentence construction, and use of modifiers. Our work is based on the analytical framework for scoring early writing proposed by Kabel et al. (2022), where each text is first annotated by a human expert according to a predefined coding scheme and subsequently scored using statistical Rasch modeling (Rasch, 1960). We investigate two different strategies for estimating these scores automatically: 1) we propose a system for identifying the central linguistic features automatically mimicking the role of the human experts and 2) we train state-of-the-art discriminative machine learning models to predict the proficiency scores directly from the texts. We conduct a number of experiments to evaluate and compare the two approaches. Our results show strong and statistically significant correlations between the scores generated using the automatic system and scores based on human experts. We also estimate and report the reliability of the individual linguistic features in the automatic annotation system. Finally, we also propose and evaluate an extension of the statistical model, which allows the model to compensate for potential systematic errors in the automatic annotations. The article thereby contributes to the area of automated essay scoring (AES) and shows that it is possible to provide teachers with automated valid and reliable knowledge about the development of their students' writing competences, which they can use in their feedback to students.

1. Introduction

Learning to write is crucial in many ways (facilitating education, enabling communication, etc.), but writing is a highly complex skill with several dimensions to it. Understanding and supporting learning trajectories of writing from the earliest years of schooling is therefore of utmost importance for young people's educational success and everyday life. A fine-grained characterization of stages of writing proficiency is an essential prerequisite for targeted and differentiated formative feedback for improved learning. However, producing high-quality feedback for writing in a common classroom setting is a very time-consuming task that teachers often find difficult to prioritize (Graham et al., 2015, Saliu-Abdulahi et al., 2017). Motivated by these challenges, the objective of this paper is to develop an automated system for fine-grained proficiency scoring. Such a system will make it possible to provide teachers with instantaneous, substantive, and detailed knowledge about what their students might currently struggle with in their development of writing competencies. Teachers can use this information in combination with other relevant knowledge to give students feedback and guide their writing. We focus on early-stage writing in Danish, whereas the general literature on automated essay scoring (AES) is dominated by English (Beigman Klebanov & Madnani, 2020, Ramesh & Sanampudi, 2022) and typically targets essays written by older students. To achieve the objective, we propose to automate the recent theoretical framework for characterizing early-stage writing by Kabel et al. (2022). The framework consists of three dimensions (text construction, sentence construction, and use of modifiers), where each dimension can be associated with a numerical proficiency scale (Bundsgaard et al., 2022) and relies on the identification of a rich set of linguistic features. Thus, evaluating the proficiency scores is a two-step process. In the first step, each text is annotated according to a predefined coding scheme and in the second step, the set of identified linguistic features is translated to proficiency scores

* Corresponding author.
E-mail addresses: miri@dtu.dk (M.R. Andersen), kabel@edu.au.dk (K. Kabel), JBRE@kp.dk (J. Bremholm), jebu@edu.au.dk (J. Bundsgaard), lkai@dtu.dk (L.K. Hansen).

https://doi.org/10.1016/j.caeai.2023.100168
Received 19 June 2023; Received in revised form 1 September 2023; Accepted 17 September 2023
Available online 22 September 2023

using statistical models (see Fig. 1a). This framework enables analysis and characterization of writing systematically and thus, facilitates individual and detailed formative feedback. However, the framework does require identification and annotation of a large set of linguistic features for each text, and hence, the framework is infeasible for teachers to deploy in a classroom setting as well as for large-scale corpus analysis. In this work, we study the potential of machine learning and natural language processing methods for identifying these linguistic features in an automated fashion, and subsequently study how accurately the proficiency scores can be estimated. Automatic systems for feature identification and proficiency scoring have numerous important applications and perspectives. For example, automatic proficiency scoring enables the analytical framework to be used by teachers in classroom settings for frequent and precise student-specific formative feedback as well as enables characterization and modeling of individual student learning trajectories. Automatic and reliable identification of linguistic features also makes large-scale corpus analysis feasible, which may otherwise be prohibited by the cost of manual annotation. Finally, an automatic scoring system also has the potential to reduce common human biases such as rater's fatigue and drift, stereotyping, halo effects, and inconsistencies (Taghipour, 2017). The explicit representation of linguistic features in the framework promotes both interpretability and explainability of the predicted scores. However, identifying all features may constitute a statistically harder problem than estimating the proficiency scores directly using discriminative machine learning methods. Moreover, Transformer-based machine learning models have recently demonstrated superior predictive performance across a wide range of natural language processing tasks (Devlin et al., 2019), but they lack direct interpretability. Therefore, we implement, evaluate, and compare both strategies for automated proficiency scoring of early-stage writing in Danish as our core contribution. For the first strategy, we implement an automatic annotation system for identifying the linguistic features automatically using state-of-the-art methods for dependency parsing and part-of-speech tagging (see Fig. 1b). Given the (approximate) feature set, it is straightforward to estimate the proficiency scores using the so-called Rasch model (Rasch, 1960). Our results show that not all features can be identified with the same level of accuracy. As a secondary contribution, we propose an extension of the Rasch model, which accounts for such systematic biases. For the second strategy, we train a Transformer-based model to predict the proficiency scores using the texts directly, completely omitting the need for explicit feature identification (see Fig. 1c). We test, evaluate and compare both methods using the data set used in Bundsgaard et al. (2022).

Our research questions (RQ) are summarized in the following:

1) How reliably can we identify the individual linguistic features of the scoring framework?
2) How accurately can the scoring framework by Kabel et al. (2022) be automated using machine learning?
3) Can the scoring accuracy be improved by omitting the need for explicit feature representation using purely discriminative machine learning models?

2. Literature review

Our work is closely related to the field of *automated essay scoring* (AES), but the focus of AES research tends to be on automated methods for summative feedback and grading, while we focus on methods for formative feedback. The focus of our work is early writing (ages 6-8) in Danish, whereas the majority of work on AES focused on more senior students and are dominated by English (Ramesh & Sanampudi, 2022). For example, Lee et al. (2023) used AES to provide formative feedback for a large online college course. There does exist work on other languages, e.g. Johan Berggren et al. (2019) and Horbach et al. (2017) constructed AES systems for Norwegian and German, respectively. Furthermore, an automated tool for text analysis is developed for Swedish and English (Palmér, 2018). Another automated tool for English is CohMetrix (McNamara et al., 2014); however, whereas these tools support analysis and assessment of student text, automated assessment is not build-in. There is little or no prior work on AES for Danish. Lorenzen et al. (2019) conducted a large-scale study of writing style on more than 100 K texts written by high-school students in Denmark. Using Siamese neural networks they constructed a text similarity metric for comparing students' newer writing with their first writing in high-school and perform a clustering analysis based on the similarities. While the authors do hypothesize that some of the clusters may help reveal at-risk students, both the student population and focus in their study differ substantially from our objective.

Modern AES systems, in general, are based on machine learning and statistics (Ramesh & Sanampudi, 2022). Deep neural networks designed for sequential modeling have also been used for AES. For example, Alikaniotis et al. (2016) investigated the use of recurrent neural networks (RNNs) and long short-term memory (LSTM) networks in AES. The authors trained LSTMs with different word embeddings Mikolov et al. (2013) on a dataset of 12976 texts written by grades 7 to grade 10 students. Their best model exhibited a Spearman's correlation coefficient of 0.91 between the predicted and ground truth scores and it outperformed models based on manual feature engineering in terms of predictive performance. Several studies also deployed Transformer-architectures like BERT for AES (Devlin et al., 2019, Wang et al., 2022, Ludwig et al., 2021). Mayfield and Black (2020) observed that such can exhibit strong predictive performance, but they are computationally expensive, requires a significant amount of data, and the cost is not always justified when compared to simpler baselines. In contrast to classical AES methods driven by feature engineering, modern approaches based on deep networks allow more flexible mappings and enable end-to-end learning of AES models, obviating the need for handcrafted features. However, a common problem with neural approaches is that they often require significantly more data to generalize well and such large datasets may not exist for the AES task at hand (Ramesh & Sanampudi, 2022). Moreover, Boulanger and Kumar (2018) trained a non-linear regression model based on deep neural networks to predict the holistic scores of 722 essays. Each essay was summarized using 1463 automatically extracted linguistic features, which was reduced to 96 features due to lack of generalization and the authors concluded that they need more labeled data to improve model generalization. Uto et al. (2020) further argue that methods based on feature engineering have an advantage in terms of interpretability and explainability, whereas methods based on neural architectures obviate the need for feature engineering and may exhibit stronger predictive power. To alleviate the "black-box" nature of neural network models, (Kumar & Boulanger, 2020) proposed to combine deep learning methods with Shapley additive explanations (Lundberg & Lee, 2017) to improve interpretability and explainability. For a more detailed review of AES in English, we refer to the recent review paper by Ramesh and Sanampudi (2022) and to the work by Beigman Klebanov and Madnani (2020) for a more historical overview.

In contrast to English, Danish is a low-resource language. According to Kirkedal et al. (2019), the availability of natural language models for Danish is rather limited compared to languages like English. We conclude this section by highlighting some noteworthy available tools for Danish: DaCy (Enevoldsen et al., 2021) is a Danish version of the popular NLP suite called SpaCy (Honnibal et al., 2019) supporting dependency parsing, named entity recognition, and sentiment analysis. DaNLP is a repository of NLP datasets and pre-trained models for Danish, which also contains pre-trained models for several NLP tasks, e.g. co-reference resolution and sentiment analysis. Stanza (Qi et al., 2020), a tool for dependency parsing based on the Universal Dependencies formalism (Nivre et al., 2017), also provides a pre-trained version for Danish based. Finally, there are also several available BERT models pre-trained for Danish through the Hugging Face repository (Wolf et al., 2020).

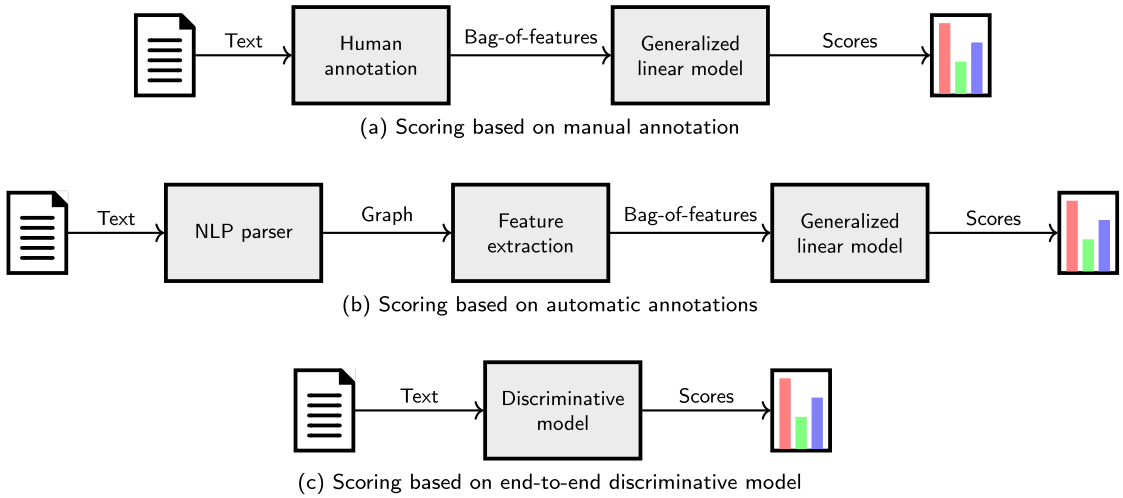


Fig. 1. Scoring pipelines: (a) scoring based on manual annotations: first a text is annotated using the coding scheme to derive the bag-of-feature representation characterizing the text. Using this representation, the proficiency with respect to each of three dimensions (sentence construction, text construction, and modifiers) is estimated using a set of statistical models (b) scoring based on automatic annotation: here the human annotated step is replaced by an NLP parser followed by a feature extraction to produce the bag-of-features representation. (c) scoring based on discriminative model: we combine a pre-trained BERT language model with a linear classifier to predict the scores directly, omitting the need for annotation.

3. Methods

In this section, we will describe the proposed methodology for implementing, evaluating, and comparing the two systems for automating the framework by Kabel et al. (2022). Proficiency scoring using the framework is a 2-step procedure (see Fig. 1(a)): first, each text is annotated by human experts using the coding scheme developed by Kabel et al. (2022). The first step effectively reduces each text to a simple bag-of-features representation (Baeza-Yates & Ribeiro-Neto, 1999). In the second step, the proficiency with respect to each of the three dimensions is estimated using statistical modeling. Section 3.1 briefly summarizes the framework, coding scheme as well as the dataset used for this study. Next, Section 3.2 describes how the features of the framework can be identified automatically using the Stanza NLP module (Qi et al., 2020) and section 3.3 outlines the details of how the annotated texts are mapped to the set of proficiency scales via statistical modeling. As with any machine learning system trained using finite datasets, the automatic annotation system is not perfect and will produce incorrect annotations for some books. Consequently, these annotation errors will propagate to the downstream statistical analysis and induce errors in the estimated proficiencies. Motivated by this observation, we propose, in Section 3.4, an extension of the statistical model which compensates for the systematic biases of the automatic annotation system to reduce the potential impact of such annotation errors. Finally, we also investigate the feasibility of estimating the proficiency scores using discriminative models (Section 3.5).

3.1. Coding scheme and dataset

We now briefly summarize the coding scheme and the dataset (Kabel et al., 2022). The coding scheme was proposed as a part of a framework for examining linguistic features of early writing and consists of both text-level features (e.g. conjunction, reference, semantic universe, genre, etc.) and sentence-level features (e.g. verbal phrases, agent composition, modifiers, etc.). The framework takes inspiration from both systemic functional linguistics (Halliday & Hasan, 1976) and a structuralist tradition for describing the order of parts of speech in sentences in the Danish language (Diderichsen, 1987, Hansen & Heltoft, 2011) and was developed through a theory- and data-driven process. It addresses current shortcomings in the literature, dominated by for example studies with a formal interest in length (Crossley, 2020); however, at the same time, it balances between an in-depth interest and the scope of the digitized student text dataset. For more details, we refer to (Kabel et al., 2022). The majority of the linguistic features in the framework are *multi-label* features and a few features are *single-label* features. For example, the sentence-level feature *agent composition* is a multi-label feature and can be assigned a subset of the following 8 labels: *ellipsis, infinitive clause, normalization, pronoun, proprium, noun, numerals, post-modifiers*. We refer to the individual labels as *items* for the given feature. The dataset consists of $N = 803$ texts written by primary-grade students (aged 6-8, with a small sub-set written by students aged 9-10), and we refer to such texts as 'books' in the following. All books were written in a digital learning platform,[1] which builds on ideas from early functional writing pedagogy (Liberg, 1990, Korsgaard et al., 2015) and is used by almost half of all compulsory schools in Denmark. Before the project began, we obtained the required authorizations and parental consent. The entire dataset corresponds to a total of 8955 sentences. Each book has been annotated by linguistic experts (three of the authors of this article, and three colleagues) using the coding scheme described above. The total number of possible items per text is 427 and is distributed among 39 features. The reliability of the annotations was ensured by double coding 15 percent of the books (inter-coder agreement of min. 80 percent for the different features) and by controlling and if necessary re-coding all books for inconsistencies detected by the double coding (Bremholm et al., 2022). We consider these annotations to be the *gold standard* and use them for training and evaluating the proposed methods using cross-validation (Stone, 1974). Using a subset of 20 of the 39 features, Bundsgaard et al. (2022) constructed numerical proficiency scales for the three dimensions based on statistical modeling. The first two columns of Table 1 show the subset of features included in the statistical analysis as well as which dimension(s) they are used in. The dataset is highly imbalanced due to some features being extremely rare, e.g. the feature *Recipient modifiers* only occurs 5 times of the total 8955 sentences. The purpose of the statistical analysis is to simultaneously estimate the *difficulty* of each item for each feature as well as the proficiency *score* of each book using a generalized linear model. Subsequently, the estimated item difficulties can be used to estimate the proficiency scores of new, unseen books. We describe the details of the

[1] www.writereader.com.

statistical modeling in Section 3.3, but first, we describe the automatic annotation system.

3.2. Automatic annotation and feature identification

The purpose of the automatic annotation system is to take a text as input and produce a bag-of-features representation. As noted, Danish is a low-resource language (Kirkedal et al., 2019) and there are no available NLP tools that allow direct identification of all relevant features and items in the coding scheme. Instead, we propose a two-step approach for the automatic annotation system, where we apply a simple feature extraction system on top of a state-of-the-art dependency parser and part-of-speech tagger (see Fig. 1b). Among the available tools, the Python NLP module called Stanza (Qi et al., 2020) is best aligned with our task. Stanza encompasses a toolset for linguistic analysis of many languages, including Danish. It uses the Universal Dependencies (UD) formalism (Nivre et al., 2017), and it supports part-of-speech tagging, syntactic dependency parsing, and named entity recognition among other features. The engine is based on modern deep neural networks and the model architecture for the dependency parser is a deep biaffine neural network based on Bidirectional Long Short-Term Memory-units (Dozat & Manning, 2017). We use models, which have been pre-trained using The Danish Universal Dependencies Treebank dataset (Johannsen et al., 2015). The output of the parser model is a directed graph representation, where the nodes correspond to words of the text and the edges of the graph represent dependency relations between the words. The nodes also contain information about part-of-speech, morphological features, and named entities. The graph representation forms the basis of the feature extraction system. Several features and items from the coding framework can be derived, either exactly or approximately, from the graph representation using simple rules. For example, the time and tense of verbal phrases can be deduced directly from the graph representation by identifying the relevant verbs and extracting the time and tense information (Nivre et al., 2017). In contrast, features involving semantic roles (e.g. *agent composition* and patient modifiers) can not be uniquely decoded from the graphs. Instead, we approximate these features using information about dependency relations, e.g. grammatical subject and object. Finally, semantic features like *process type* and *attitude* cannot be identified based on the graph alone. The feature *process type* $\in$ {relational, mental, verbal, material} can be approximately identified by first extracting the relevant verbs from the graph, and subsequently classifying them using a simple look-up dictionary approach. We implemented 17 out of the 20 features used by Bundsgaard et al. (2022). The features *Recipient composition* and *Recipient modifiers* are extremely sparse and are only observed 5 and 10 times, respectively, in the entire data set (see Table 1). Consequently, we cannot reliably evaluate the quality of the automatic annotations, and therefore, we do not implement these features. Items for the feature *reference* cannot be deduced from the dependency graph alone and are therefore not implemented in this work.

3.3. Proficiency scoring

To produce the proficiency score for a given dimension, the text is first analyzed using natural language processing, which produces a graph representation of the text. Next, the graph is fed to a feature extraction block, which converts the text into a binary bag-of-features representation. After this step, each text is characterized by a binary feature vector, where each entry indicates whether the text contains a given item or not. These representations are subsequently mapped to a set of proficiency scales (text construction, sentence construction, and modifiers) as follows. Let $y_{n,i} \in \{0,1\}$ represent the i'th item of the n'th book such that $y_{n,i} = 1$ if and only if the n'th book contains item i and $y_{n,i} = 0$ otherwise. Using matrix notation the entire dataset can be represented using a matrix $Y \in \{0,1\}^{N\times D}$, where N is the number of texts/books and D is the number of items. The data matrix Y can now be decomposed into a set of *book scores* and *item difficulties* using a generalized linear model of the form

$$p(y_{n,i} = 1|\theta_n, \delta_i) = \sigma\left(\theta_n - \delta_i\right), \tag{1}$$

where θ_n is the score for the n'th book and δ_i is the difficulty of the i'th item, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function. This model is known as the *Rasch model* (Rasch, 1960). The difference $\theta_n - \delta_i$ completely controls the probability of observing a specific item in a given book. For a fixed set of item difficulties $\{\delta_i\}_{i=1}^{D}$, a book with a greater score (i.e. larger value for θ_n) will, in expectation, contain more items compared to a book with a lower score (i.e. lower values for θ). Similarly, books with the same value of θ_n will, in expectation, contain more items with lower item difficulties compared to items with larger item difficulties. This model separates book scores from item difficulties and, thus, provides a framework for scoring books with respect to interpretable proficiency scales.

The real-valued book scores θ_n can further be mapped to discrete, ordinal quantities $z_n \in \{1,2,3,4\}$ to ease the interpretation and communication (Bundsgaard et al., 2022). The idea is to partition the real numbers $\mathbb{R}$ into four disjoint intervals corresponding to the four different levels of z_n. The intervals are constructed using order statistics as follows. Let $\theta_{(1)} \leq \theta_{(2)} \leq \ldots \leq \theta_{(N)}$ denote the ordered scores and let $\Delta = \theta_{(N)} - \theta_{(1)}$ denote their range. The map from $\theta_n \in \mathbb{R}$ to $z_n \in \{1,2,3,4\}$ is defined as follows

$$z_n = \begin{cases} 1 & \text{if} \quad \theta_n - \theta_{(1)} \leq \frac{1}{4}\Delta \\ 2 & \text{if} \quad \frac{1}{4}\Delta < \theta_n - \theta_{(1)} \leq \frac{2}{4}\Delta \\ 3 & \text{if} \quad \frac{2}{4}\Delta < \theta_n - \theta_{(1)} \leq \frac{3}{4}\Delta \\ 4 & \text{if} \quad \frac{3}{4}\Delta < \theta_n - \theta_{(1)}. \end{cases} \tag{2}$$

Each book can then be summarized using $\hat{z}_n = \arg\max_c p(z_n = c|y_n) \in \{1,2,3,4\}$.

We use Bayesian statistics (Gelman et al., 2013) to estimate the parameters of the model assuming the following joint model

$$\theta_n|\sigma_\theta \sim \mathcal{N}(0, \sigma_\theta), \tag{3}$$

$$\delta_i|\sigma_\delta \sim \mathcal{N}(0, \sigma_\delta), \tag{4}$$

$$\sigma_\theta \sim p(\sigma_\theta), \tag{5}$$

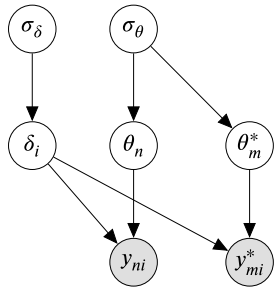
$$\sigma_\delta \sim p(\sigma_\delta), \tag{6}$$

where $p(\sigma_\theta)$ and $p(\sigma_\delta)$ are prior distribution for the prior scale of the book scores and item difficulties, respectively. Following the work of Bundsgaard et al. (2022), we model each proficiency score for each dimension independently, but we note that a hierarchical formulation might lead to more robust inference (Gelman et al., 2013).

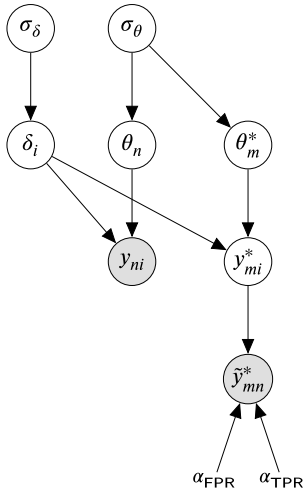
We implement all models in the probabilistic programming language Stan (Stan Development Team, 2012), which uses Markov Chain Monte Carlo (MCMC) methods to estimate the posterior distributions. For all experiments, we use the dynamic Hamiltonian Monte Carlo (Hoffman & Gelman, 2014) algorithm using 4 MCMC chains, 2000 iterations, 1000 warm-up samples, and thinning of 2. Moreover, we use $p(\sigma_\delta) = \delta(\sigma_\delta - 3)$ and $p(\sigma_\theta) = \text{Exp}(\lambda)$ with $\lambda = 0.2$ for all experiments. We ensure that the potential scale reduction factor is below 1.1 to avoid mixing problems (Gelman et al., 2013). We use the posterior mean as estimators for the parameters θ_n and δ_i.

3.4. Compensating for systematic errors

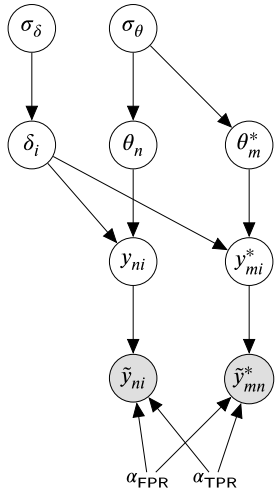
The model described in the previous section implicitly assumes that the observations Y are 'noiseless'. However, the automatic annotations system is not perfect and will produce errors, where the error rates depend on the specific item. In the following, we propose an extension of the Rasch model that allows the model to compensate for systematic errors in automatic annotations.



(a) $\mathcal{M}_1$: Classic Rasch model (assuming noiseless observations)



(b) $\mathcal{M}_2$: Rasch model assuming noisy observation at test time



(c) $\mathcal{M}_3$: Rasch model assuming noisy observation at both training and test time

Fig. 2. (a) Graphical model representation of the Rasch model, when estimating the model using y (training set) and making predictions using for y^* (test set). (b) Graphical model when making predictions using noisy observations $\tilde{y}^*$. (c) Graphical model when both training and test observations are assumed to be corrupted by noise. Note: Plates in all three graphical models are not shown. Parameters α_{FPR} and α_{TPR} represents the false positive rate and true positive rate, respectively, for each item.

Let $y_{ni} \in \{0, 1\}$ denote the gold standard annotation and the $\tilde{y} \in \{0, 1\}$ denote the automatic annotation, then the *true positive rate* (TPR) and the *false positive rate* (FPR) for the i'th item are defined as

$$\alpha_{\text{TPR},i} = p(\tilde{y}_{ni} = 1 | y_{ni} = 1), \tag{7}$$

$$\alpha_{\text{FPR},i} = p(\tilde{y}_{ni} = 1 | y_{ni} = 0). \tag{8}$$

These assumptions can be combined in the observation model for $\tilde{y}_{ni}$

$$p(\tilde{y}_{ni} = 1 | y_{ni}) = \begin{cases} \alpha_{\text{TPR},i} & \text{if} \quad y_{ni} = 1 \\ \alpha_{\text{FPR},i} & \text{if} \quad y_{ni} = 0. \end{cases} \tag{9}$$

Marginalizing out y_{ni} using the distribution in eq. (1) yields a likelihood for noisy annotations,

$$\begin{aligned} p(\tilde{y}_{ni} | \theta_n, \delta_i) &= \alpha_{\text{FPR}} \left[1 - \sigma(\theta_n - \delta_i)\right] + \alpha_{\text{TPR}} \cdot \sigma(\theta_n - \delta_i) \\ &= \alpha_{\text{FPR}} + \sigma(\theta_n - \delta_i) \left[\alpha_{\text{TPR}} - \alpha_{\text{FPR}}\right]. \end{aligned} \tag{10}$$

This construction can be seen as a generalization of the Rasch model. That is, if the automatic annotation system identifies the i'th item perfectly, i.e. $\alpha_{\text{TPR}_i} = 1$ and $\alpha_{\text{FPR}_i} = 0$, then the likelihood in eq. (10) reverts to the likelihood of the Rasch model in eq. (1). In the other extreme, where $\alpha_{\text{FPR}_i} = \alpha_{\text{TPR}_i}$, the model effectively ignores the observations for the i'th item. The proposed model interpolates between these extremes depending on the error rates of the individual items. The error rates α_{TPR_i} and α_{FPR_i} can be estimated using training data. Since the dataset is rather small ($N = 803$) and due to the severe sparsity of several items, we apply Laplace smoothing (with smoothing parameter $\lambda = 1$) to estimate these probabilities (Manning et al., 2008).

3.5. Estimating proficiency scores using discriminative modeling

Identifying the large set of items may constitute a statistically harder problem compared to estimating the proficiency scores directly using discriminative modeling. Thus, it may be possible to achieve comparable or better predictive performance using discriminative machine learning models. However, the trade-off may be reduced interpretability compared to the two-step of approach feature identification and subsequent scoring (see Fig. 1). To investigate this hypothesis, we employ a state-of-the-art language model and fine-tune it for classifying the discrete level z_n for each dimension. Specifically, we use a BERT model (Devlin et al., 2019) with pre-trained weights[2] from Hugging Face (Wolf et al., 2020). We fine-tune the BERT model individually for each dimension using 3 epochs, a batch size of 8, and weight decay fixed to 10^{-2}. As a baseline, we also fit ordinal regression models based on the logarithm of the word counts using the Python package *mord* (Pedregosa-Izquierdo, 2015).

4. Experiments

We designed and conducted several experiments to investigate the performance and behavior of the automated scoring pipelines (see Fig. 1). In Section 4.1, we address RQ1, by evaluating the performance of the automatic annotation system (AAS), and in Section 4.2, we quantify the accuracy and reliability of proficiency scores estimated using the AAS addressing RQ2. In Section 4.3, we investigate whether the performance of the system can be improved by accounting for the limitations and biases of the AAS. Finally, in Section 4.4 we investigate how well the proficiency scores can be predicted using discriminative models addressing RQ3.

4.1. Experiment 1: reliability of feature identification using the automatic annotation system

The purpose of this experiment is to assess and quantify the feature identification performance of the AAS. First, we use the AAS to re-annotate the entire dataset, and then we compare the automatic annotations with the human annotation, which we consider to be the gold

[2] We use the pre-trained model with the identifier: `Maltehb/danish-bert-botxo`.

Table 1
Results for experiment 1: Performance of automatic annotation system. The rows in the first section represent sentence-level features and the rows in the second section represent text-level features. The column *scales* indicates which of the proficiency scales use the features (S: Sentence construction, T: Text construction, and M: Modifiers). The columns *count* and *labels* denote the total number of occurrences of the feature and the number of possible items per feature, respectively. The values for precision, recall, and F_1 scores are weighted averages with respect to counts.

Feature	Dimensions	Count	Labels	Accuracy	Precision	Recall	F_1
Agent modifiers	MS	697	13	0.99	0.66	0.64	0.64
Agent composition	S	488	4	0.99	0.63	0.63	0.62
Free adverbial phrases and clauses (construction)	S	4384	7	0.94	0.57	0.65	0.60
Fixed adverbial phrases	ST	199	1	0.99	0.70	0.53	0.61
Sentence opener	S	7360	5	0.95	0.89	0.75	0.81
Free adverbial phrases	MST	4307	8	0.95	0.68	0.53	0.58
Patient modifiers	MS	505	13	0.99	0.35	0.51	0.41
Patient composition	S	734	4	0.99	0.51	0.44	0.46
Recipient modifiers	MS	10	4	-	-	-	-
Recipient composition	S	5	1	-	-	-	-
Predicate modifers	MS	1767	14	0.99	0.54	0.52	0.48
Predicate composition	S	1445	5	0.99	0.67	0.71	0.67
Verbal phrase (tense)	S	918	3	1.00	0.82	0.77	0.80
Process type	S	7671	4	0.95	0.89	0.87	0.88
Verbal phrase (time)	S	7126	3	0.97	0.87	0.87	0.87
Attitude	T	623	3	0.77	0.80	0.40	0.34
Conjuction	T	586	5	0.90	0.68	0.86	0.74
Interjections	T	165	2	0.83	0.25	0.31	0.28
Dialogue/voices	T	176	1	0.94	0.72	0.36	0.48
Reference	T	382	1	-	-	-	-

standard. For each item and each feature, we compute the following metrics. The classification accuracy is defined by

$$\mathcal{A} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}}, \tag{11}$$

where tp, tn, fp, and fn denotes the number of true positives, true negatives, false positives, and false negatives, respectively. Due to the severe imbalance of the dataset, we also compute the precision, recall, and the F_1 score:

$$\mathcal{P} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \approx p(y = 1 | \tilde{y} = 1), \tag{12}$$

$$\mathcal{R} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} \approx p(\tilde{y} = 1 | y = 1), \tag{13}$$

$$F_1 = 2\frac{\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{14}$$

where $y \in \{0, 1\}$ is the human annotation for the given item and $\tilde{y} \in \{0, 1\}$ denotes the corresponding automatic annotation. The precision $\mathcal{P}$ measures the fraction of the identifications that are true positives, recall $\mathcal{R}$ measures the fraction of true cases detected, and the F_1-score is the harmonic mean of precision and recall (Rijsbergen, 1979). For multi-label features, all metrics are evaluated for each possible label for each feature, but we summarize the performance for each feature by computing a weighted average of the metrics with respect to the relative frequencies of each label. The results are summarized in Table 1. The results show that the F_1-score varies greatly from feature to feature and features of a more grammatical nature, such as sentence opener and verbal phrase, are relatively easy to extract from the dependency graphs and part-of-speech information, and therefore show higher F_1 scores. In contrast, the interjection feature has the lowest $F_1 \approx 0.28$ among all features. Moreover, items related to agent modifiers and composition are generally more reliably identified compared to items related to patient modifiers and composition. Moreover, we note that there are three sources of errors: 1) Stanza predicts an erroneous graph representation of the text, 2) the subsequent feature extraction fails, or 3) there might be errors in the human annotations. However, we do not distinguish between the three modes in this evaluation.

4.2. Experiment 2: reliability of estimated proficiency scoring using automatic feature identification

In this experiment, we investigate how the biases and limitations of the AAS affect the downstream performance on the estimated proficiency scores. To quantify this, we first estimate the parameters of the binary Rasch model described in Section 3.3 based on the manually annotated dataset. Treating these parameters estimates as gold standard, we can now fit the same model again, but now using the annotations from AAS to compare the results. Using the posterior mean values as parameter estimates, we compute the Spearman correlation between the (posterior mean) score parameters obtained using the human-annotated dataset $\{\theta_n\}_{n=1}^N$ and the (posterior mean) score parameters obtained using the AAS system $\{\hat{\theta}_n\}_{n=1}^N$, where $N = 803$ denotes the number of texts. We also compute the best linear fit between the two sets of parameters using ordinary least squares. We finally compute the corresponding quantity based on $\{\delta_i\}_{i=1}^D$ and $\{\hat{\delta}_i\}_{i=1}^D$, the item difficulty parameters estimated using human-annotated data and automatically annotated data, respectively. Here D is the number of items. The results are summarized in Table 2. The results show that there are strong and statistically significant correlations between the scores based on human and automatic annotations, respectively. It is also seen that the *text construction* dimension exhibits the weakest correlation among the three dimensions, but yet the estimated correlation coefficient $\rho \approx 0.75$ is large. The *sentence construction* dimension, on the other hand, shows a score correlation coefficient of $\rho \approx 0.95$. The gap is consistent with the results in Table 1 showing generally higher F_1 scores for features related to sentence construction compared to features related to text construction. We see a similar pattern for the item difficulty parameters, where Table 2 shows strong and statistically significant correlations between items for modifiers and sentence construction, whereas the item correlation for the text construction dimension is weaker and not statistically significant.

4.3. Experiment 3: accounting for the limitations of automatic annotation system in a predictive setting

A large part of the motivation behind this work is to enable automated scoring of early writing. In a practical setting, this involves a *training* phase, where the item difficulty parameters of the models are estimated using an "offline" *training set*, and then a *test* phase, where

Table 2
Results for experiment 2: Reliability of proficiency scoring using the automatic annotation system. First, we estimate the Rasch model using human annotated texts and afterwards using automatic annotated texts. Treating the parameter estimates based on human annotations as the gold standard, we compare the Spearman correlation and the ordinary least squares fit between the score parameters estimated using the two datasets. We evaluate the same metric for the difficulty parameters. The asterisk $*$ indicates that p-value < 0.0001 yielding statistical significance at $\alpha = 0.05$.

Dimension	Score correlation	Score regression	Item correlation	Item regression
Text construction	0.75*	$y = 1.07x - 0.02$	0.19	$y = 0.18x + 3.31$
Sentence construction	0.95*	$y = 0.95x + 0.01$	0.95*	$y = 1.24x - 0.36$
Modifiers	0.86*	$y = 0.91x - 0.01$	0.70*	$y = 1.08x + 0.06$

Table 3
Results for experiment 3: Using 20-fold cross-validation, we simulate the practical use case of automatic scoring and compare the performance of the three different models $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$. Model $\mathcal{M}_1$ is estimated using the human-annotated dataset, and model $\mathcal{M}_3$ is estimated using the automatic annotation, whereas model $\mathcal{M}_2$ is estimated twice: once for each of the two types of annotations.

Dimension	Training set	Model	Score correlation	Item correlation
Text construction	Human annotations	$\mathcal{M}_1$	0.73 (± 0.03)	0.99 (± 0.00)
		$\mathcal{M}_2$	0.77 (± 0.03)	0.99 (± 0.00)
	Automatic annotations	$\mathcal{M}_2$	0.76 (± 0.02)	0.17 (± 0.01)
		$\mathcal{M}_3$	0.76 (± 0.02)	0.85 (± 0.01)
Sentence construction	Human annotations	$\mathcal{M}_1$	0.94 (± 0.01)	1.00 (± 0.00)
		$\mathcal{M}_2$	0.94 (± 0.02)	1.00 (± 0.00)
	Automatic annotations	$\mathcal{M}_2$	0.94 (± 0.02)	0.94 (± 0.00)
		$\mathcal{M}_3$	0.94 (± 0.02)	0.98 (± 0.00)
Modifiers	Human annotations	$\mathcal{M}_1$	0.83 (± 0.03)	0.98 (± 0.00)
		$\mathcal{M}_2$	0.86 (± 0.02)	1.00 (± 0.00)
	Automatic annotations	$\mathcal{M}_2$	0.85 (± 0.02)	0.71 (± 0.01)
		$\mathcal{M}_3$	0.85 (± 0.02)	0.89 (± 0.00)

the estimated parameters are used for scoring new, unseen books in the so-called *test set*. In other words, the correlations in Experiment 2 are estimated in an "in-distribution" setting, but in a more practical setting, we expect some books to be "out-of-distribution" leading to greater variance of the estimated scores. In this experiment, we mimic this set-up using K-fold cross-validation to assess how well the system generalizes to unseen books (Stone, 1974). Due to the sparse nature of several items, we use $K = 20$ folds to estimate the generalization capabilities. We assume that the human annotations are available for the training sets and can be used when estimating the item difficulty parameters, whereas we assume that only automatic annotations are available for the test sets mimicking a practical application. However, we use the human annotations on the test sets to estimate the gold standard scores for reference. We evaluate and compare the three models $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ (see Fig. 2) and investigate four settings for each dimension:

1. Use model $\mathcal{M}_1$ and estimate model parameters using human annotations as training set, estimate scores for test set using automatic (AAS) annotations
2. Use model $\mathcal{M}_2$ and estimate model parameters using human annotations as training set, estimate scores for test set using automatic annotations
3. Use model $\mathcal{M}_2$ and estimate model parameters using automatic annotations (AAS) as training set, estimate scores for test set using automatic annotations
4. Use model $\mathcal{M}_3$ and estimate model parameters using automatic annotations AAS annotations as training set, estimate scores for test set using AAS annotations

Fitting model $\mathcal{M}_2$ and $\mathcal{M}_3$ further require estimates of the true positive rate α_{TPR} and false positive rate α_{FPR} for each item. For each cross-validation split, we use the training set to estimate these error probabilities. The results are summarized in Table 3. It is evident from Table 3 that the estimated score correlations for the Rasch model ($\mathcal{M}_1$) are generally lower compared to Experiment 2, e.g. for the *text construction dimension* the score correlation is decreased from 0.75 to 0.73 ± 0.03 and similar for the two other dimensions. It is also seen that the score correlations for $\mathcal{M}_2$ are identical to the $\mathcal{M}_1$ model for the sentence construction, whereas the score correlations for modifiers and text construction are slightly higher for $\mathcal{M}_2$. Interestingly, if we use automatic annotations not only for predicting the scores for books in the test sets, but also for training the models, then accounting for the limitations of the parser dramatically improves the correlation between item difficulties for all three dimensions, e.g. the item correlations for modifiers and text construction are improved from 0.71 ± 0.01 to 0.89 ± 0.01 and from 0.17 ± 0.01 to 0.85 ± 0.01, respectively. As the performance of the automatic annotation system depends heavily on the specific item, it is natural to investigate whether the performance of the system could be improved by only using items with higher accuracy and ignoring the rest. We designed the next experiment to investigate this hypothesis. Since dimensions with D items contain 2^D possible subsets, it is infeasible to exhaust the complete search space. Instead, we use a simple forward selection scheme, where item inclusion order is determined by the relative F_1-scores measured on the training sets. The results are visualized in Fig. 3, where the first panel shows the estimated F_1-scores for each included item, the second panel shows the score correlation as a function of the number of items included and the third panel shows the absolute difference between the scores scaled by the posterior standard deviation of the model with automatic annotations, i.e. $d_i = \frac{|\mu_i - \hat{\mu}_i|}{\hat{\sigma}_i}$. Fig. 3b shows the score correlations as a function of the number of included items. For all dimensions, it is seen that the number of included items can be significantly reduced without sacrificing accuracy, e.g., for sentence construction approximately 20 items are sufficient to reach full accuracy. Including the noisiest items for text construction and modifiers seem to decrease the score correlation a bit for $\mathcal{M}_1$ model, whereas $\mathcal{M}_2$ is able to compensate due to the information about item reliabilities. The experiment also highlights another important issue: the Rasch model ($\mathcal{M}_1$) leads to overconfident uncertainty estimates as the number of included items is increased. This is evidenced in Fig. 3c, which shows the absolute difference between the score estimates divided by the posterior standard deviation for the

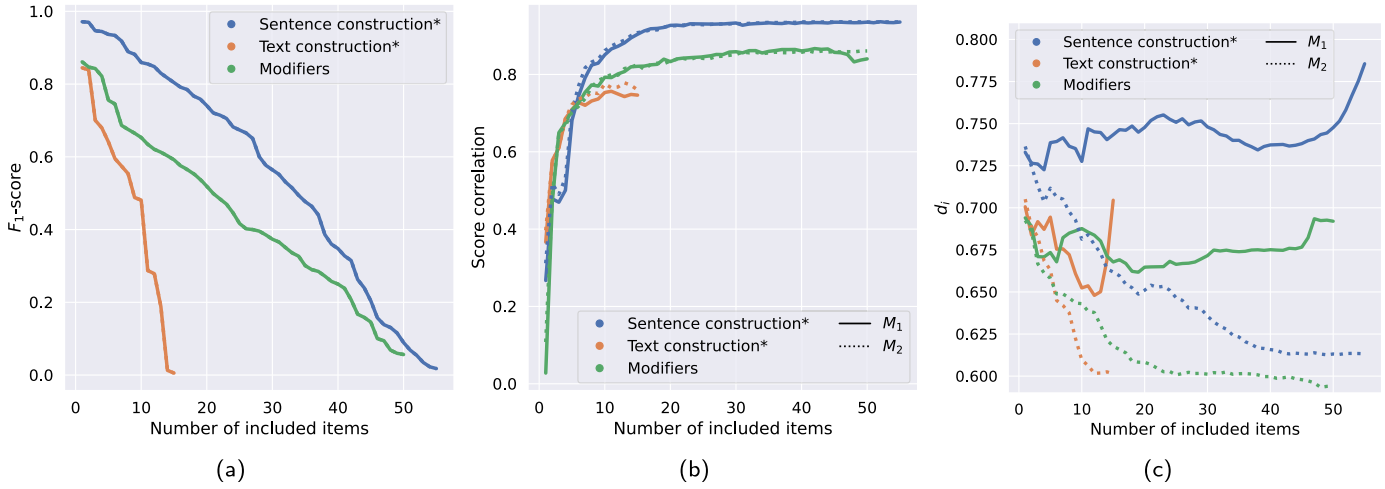


Fig. 3. Results for experiment 3: Reliability of proficiency scales for increasing subsets of items. Items are included in a forward selection manner based on the F_1-score in the training set. (a) shows the F_1-score (averaged over cross-validation folds) vs. the number of included items. (b) shows the average Spearman correlation between the posterior means of $\{\theta_n\}_{n=1}^N$ and $\{\hat{\theta}\}_{n=1}^N$ for models $\mathcal{M}_1$ and $\mathcal{M}_2$. (c) shows the (averaged) scaled absolute difference $d_i = \frac{|\mu_{\theta_i} - \hat{\mu}_{\theta_i}|}{\hat{\sigma}_{\theta_i}}$, where $\hat{\sigma}_{\theta_i}$ denotes the standard deviation of the posterior distribution of $\hat{\theta}_i$. Figure best viewed in color.

Table 4
Results for experiment 4: Results for estimating the proficiency scores (1-4) for each of the three dimensions using 20-fold cross-validation. We report the results using the quadratic weight kappa (QWK) and classification accuracy.

Dimension	Method	QWK	Accuracy
Text construction	Majority class	0.00 (± 0.00)	0.40 (± 0.03)
	Ordinal regression using word counts	0.52 (± 0.04)	0.53 (± 0.03)
	Classification using BERT	0.65 (± 0.05)	0.59 (± 0.04)
	$\mathcal{M}_1$ (human annotations)	0.70 (± 0.03)	**0.60 (± 0.03)**
	$\mathcal{M}_2$ (human annotations)	**0.73 (± 0.03)**	0.57 (± 0.04)
	$\mathcal{M}_3$ (automatic annotations)	0.56 (± 0.04)	0.33 (± 0.04)
Sentence construction	Majority class	0.00 (± 0.00)	0.54 (± 0.03)
	Ordinal regression using word counts	0.69 (± 0.04)	0.76 (± 0.02)
	Classification using BERT	0.74 (± 0.06)	0.80 (± 0.04)
	$\mathcal{M}_1$ (human annotations)	**0.84 (± 0.03)**	**0.86 (± 0.03)**
	$\mathcal{M}_2$ (human annotations)	0.82 (± 0.03)	0.80 (± 0.03)
	$\mathcal{M}_3$ (automatic annotations)	0.74 (± 0.03)	0.62 (± 0.03)
Modifiers	Majority class	0.00 (± 0.00)	0.49 (± 0.05)
	Ordinal regression using word counts	0.69 (± 0.05)	**0.75 (± 0.03)**
	Classification using BERT	0.69 (± 0.05)	0.74 (± 0.03)
	$\mathcal{M}_1$ (human annotations)	0.73 (± 0.03)	0.74 (± 0.03)
	$\mathcal{M}_2$ (human annotations)	**0.76 (± 0.03)**	0.74 (± 0.02)
	$\mathcal{M}_3$ (automatic annotations)	0.66 (± 0.03)	0.55 (± 0.03)

model with automatic annotations, i.e. $d_i = \frac{|\mu_{\theta_i} - \hat{\mu}_{\theta_i}|}{\hat{\sigma}_{\theta_i}}$. This metric increases for $\mathcal{M}_1$ because the models decrease their uncertainty estimates without improving the accuracy. On the other hand, the $\mathcal{M}_2$ models do not increase their confidence when including items that contain more noise than signal.

4.4. Experiment 4: estimating the proficiency scores using discriminative modeling

This experiment is designed to investigate to what degree it is possible to predict the discretized proficiency scores z_n using discriminative models, and thus, alleviating the need for the two-step approach of annotation followed by scoring as for $\mathcal{M}_1, \mathcal{M}_2$, and $\mathcal{M}_3$. To quantify the performance, we measure and report the classification accuracy as well as the quadratic weighted Cohen's kappa (QWK) metric (McHugh, 2012) to capture the ordinal nature of z_n. Again, we rely on K-fold cross-validation with $K = 20$ to estimate the generalization of the models. As an additional baseline, we also report results for a simple classifier that always predicts the majority class. The results are summarized in Table 4. Several interesting observations can be made from the results. First, all models showed significantly better predictive performance with respect to both metrics compared to simply always predicting the class with the largest frequency. Second, $\mathcal{M}_1$ and $\mathcal{M}_2$ are always comparable or better than both discriminative approaches (i.e. ordinal regression and the pre-trained BERT classifier) with respect to both metrics. Third, the performance of BERT is comparable to or better than the simple regression model.

5. Discussion

We implemented two strategies for automating the theoretical analysis framework by Kabel et al. (2022). To evaluate and compare the two strategies and to answer our research questions, we designed and conducted four experiments. The experiment (see Section 4.1) was designed to answer the first research question directly, i.e. how reliable can the individual features of the framework be identified automatically? Overall, the results showed that the identification reliability, as quantified by the F_1-metric varied greatly from feature to feature and that features related to grammar can generally be identified more reliably compared to features of a more semantic nature. The results from

this experiment may also be of interest to adjacent fields involving components of NLP and computational linguistics in Danish, e.g. analysis of writing style etc. Experiments 2 & 3 (see Sections 4.2 and 4.3) targeted the second research question, i.e. how accurate can the framework be automated using machine learning? Overall, the results showed strong and statistically significant Spearman's correlations between the scores computed using the gold standard annotations and scores computed using the automatic annotations for all dimensions. These strong associations are already promising for the applications, but we want to emphasize that these correlations are based on scoring individual short books (median is 24 words per book). Therefore, we hypothesize that aggregating the scores across multiple books might lead to even better estimators. The models and methods developed in this work may also have applications beyond this work. For example, the framework by Kabel et al. (2022) was developed by based on manual annotation of $N = 803$ books. However, our work enables a large-scale analysis of an unlabeled corpus beyond the size of manual annotation and this may lead to an even more fine-grained model of writing. Evidenced by the significant increase in item correlations in Table 3, the proposed extension of the Rasch model may lead to much more accurate conclusions for such large-scale studies. We addressed the last research question in experiment 4 (see Section 4.4) by training and evaluating end-to-end models (a simple regression model and a state-of-the-art BERT model) of the proficiency scores for each dimension. The results for the regression model showed surprisingly strong predictive performance given its simplicity and very low computational cost compared to BERT, which is consistent with the observations made by Mayfield and Black (2020). The positive associations between word count and writing proficiency are not surprising (Crossley, 2020), but it is only a correlation and not a causal relationship. That is, excellent texts written by 6-8 years old are often relatively long, but long texts are not necessarily excellent. Consequently, such a model is of limited use for the purpose of providing high-quality and detailed feedback due to its relatively low degree of explainability. The predictive performances of the BERT models are relatively close to the performance of the $\mathcal{M}_1$ and $\mathcal{M}_2$ models, and it is likely that the predictive performance of BERT could be further improved by pre-training on more domain-specific data. However, despite recent advances in the field of explainability (Kumar & Boulanger, 2020), predictions from BERT models are not easily interpretable with the same level of granularity as the coding framework by Kabel et al. (2022) and will therefore be of significantly lower practical value for the purpose of the formative feedback and teachers' planning of the writing curriculum. In contrast, the identified features and items from the $\mathcal{M}_1$ and $\mathcal{M}_2$ models may directly be of great pedagogical value. However, the potential pedagogical advantages of these models are restricted and complicated by the fact that sets of identified features are noisy. As we see it, trying to fuse these two methodological approaches constitutes a very interesting endeavor for future research that would hold the promise of combining the best from both worlds.

Finally, recalling the initial objective behind this study it is pertinent to point out that, in this paper, we have demonstrated that we have been able to develop an automated system for fine-grained proficiency scoring of students' early writing (cf. the discussion above on the first and second research question). Just as important, it is worth emphasizing the strong pedagogical value of such a system. The automated scoring holds the potential of relieving the writing teachers from the demanding and time-consuming task of manually assessing texts from a whole class of students, and thus, supporting the teachers both in giving individual students well-founded and specific feedback on their writing development and in preparing the writing instruction for the whole class going forward. In this sense, automated scoring is essential in realizing the pedagogical potential of both the theoretical framework (Kabel et al., 2022) and the proficiency scales for early-stage writing (Bremholm et al., 2022) in actual, everyday writing classrooms. In light of the pedagogical benefits of automated scoring of student texts, it would be of much interest and relevance for future research to develop automated scoring systems for texts written by older students, e.g. students in middle school or lower secondary school. Such a study would also include developing an adequate theoretical framework as well as proficiency scales for the developmental trajectories of this age group. All of these aspects would be highly challenging given the fact that student texts at these later stages are more varied and complex, both linguistically and with regard to genre and disciplinary specificity, but the benefits of an automated scoring system would be equally significant for the teachers as support for a pedagogical task that today is often a neglected aspect of their instructional practice (Graham et al., 2015, Saliu-Abdulahi et al., 2017).

6. Conclusion

We implemented and evaluated two strategies for automated proficiency scoring of early-stage writing in Danish. The results showed strong and statistically significant correlations between the scores based on automatic annotations and human annotations. Specifically, writing proficiency with respect to sentence construction can be identified with very high accuracy (correlation coefficient of 0.95), whereas proficiency with respect to text construction can be identified with slightly lower accuracy (correlation coefficient of 0.75). We also demonstrated that the correlation coefficients between item parameters can be significantly improved by extending the Rasch model to capture the individual reliability and uncertainty of the items. Finally, we also trained discriminative models for predicting the proficiency scores directly and showed that they yield competitive predictive performance, but lack transparency and easy explainability.

In summary, our research shows that we can reliably identify writing proficiency automatically and this opens up several new research directions on how to incorporate automatic and fine-grained feedback to facilitate learning. In this study, we focused on automated assessment for individual student texts. However, individual texts may not be representative of students' true writing capabilities, e.g. a student may be unfocused, tired, or disturbed when writing. This limitation can be alleviated by analyzing sequences of texts over time and developing statistical models for the learning dynamics. We consider this as important future work, as this not only enables more robust statistical inferences but also allows estimating the temporal development of writing proficiency on both student and population levels. This is also likely to further improve the alignment between human and automated scores. Furthermore, our current approach does not account for how specific writing instructions may affect the realization of the individual texts. Therefore, studying the effects of writing instruction and developing the corresponding conditional statistical models are also very interesting avenues for future work.

7. Acknowlegdement

We acknowledge funding from Innovation Fund Denmark (grant number 8057-00036A). Furthermore, we would also like to thank Janus Madsen from WriteReader for all his help during the project.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 715–725). Berlin, Germany: Association for Computational Linguistics.

Baeza-Yates, R. A., & Ribeiro-Neto, B. A. (1999). *Modern information retrieval*. ACM Press / Addison-Wesley.

Beigman Klebanov, B., & Madnani, N. (2020). Automated evaluation of writing – 50 years and counting. In *Proceedings of the 58th annual meeting of the association for computational linguistics, association for computational linguistics* (pp. 7796–7810), Online, https://aclanthology.org/2020.acl-main.697.

Boulanger, D., & Kumar, V. (2018). Deep learning in automated essay scoring. In *International conference on intelligent tutoring systems* (pp. 294–299). Springer.

Bremholm, J., Bundsgaard, J., & Kabel, K. (2022). Proficiency scales for early writing development. *Writing and Pedagogy*, *13*, 121–154.

Bundsgaard, J., Kabel, K., & Bremholm, J. (2022). Validating scales for the early development of writing proficiency. *Writing and Pedagogy*, *13*, 89–120. https://doi.org/10.1558/wap.21491.

Crossley, S. A. (2020). Linguistic features in writing quality and development: An overview. *Journal of Writing Research*, *11*, 415–443. https://doi.org/10.17239/jowr-2020.11.03.01.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics.

Diderichsen, P. (1987). Elementær dansk grammatik [Elementary Danish grammar]. Gyldendal.

Dozat, T., & Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. In *5th international conference on learning representations*.

Enevoldsen, K., Hansen, L., & Nielbo, K. (2021). Dacy: A unified framework for Danish nlp.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian data analysis (3rd ed.)

Graham, S., Hebert, M., & Harris, K. (2015). Formative assessment and writing: A meta-analysis. *The Elementary School Journal*, *115*, 523–547. Publisher Copyright: © 2015, the University of Chicago. All rights reserved.

Halliday, M. A. K., & Hasan, R. (1976). *Cohesion in English*. London: Longman.

Hansen, E., & Heltoft, L. (2011). Grammatik over det danske sprog [Danish language grammar]. In *Det danske sprogog litteraturselskab*.

Hoffman, M. D., & Gelman, A. (2014). The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623.

Honnibal, M., Montani, I., Honnibal, M., Peters, H., Landeghem, S. V., Samsonov, M., Geovedi, J., Regan, J., Orosz, G., Kristiansen, S. L., McCann, P. O., Altinok, D., Roman, Howard, G., Bozek, S., Bot, E., Amery, M., Phatthiyaphaibun, W., Vogelsang, L. U., ... Patel, A. (2019). explosion/spaCy: v2.1.7: Improved evaluation, better language factories and bug fixes. https://doi.org/10.5281/zenodo.3358113.

Horbach, A., Scholten-Akoun, D., Ding, Y., & Zesch, T. (2017). Fine-grained essay scoring of a complex writing task for native speakers. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications* (pp. 357–366). Copenhagen, Denmark: Association for Computational Linguistics.

Johan Berggren, S., Rama, T., & Øvrelid, L. (2019). Regression or classification? Automated essay scoring for Norwegian. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications* (pp. 92–102). Florence, Italy: Association for Computational Linguistics.

Johannsen, A., Alonso, H. M., & Plank, B. (2015). Universal dependencies for Danish. In *International workshop on treebanks and linguistic theories (TLT14)* (p. 157).

Kabel, K., Bremholm, J., & Bundsgaard, J. (2022). A framework for identifying early writing development. *Writing and Pedagogy*, *13*, 51–87. https://doi.org/10.1558/wap.21467.

Kirkedal, A., Plank, B., Derczynski, L., & Schluter, N. (2019). The lacunae of Danish natural language processing. In *Proceedings of the 22nd nordic conference on computational linguistics* (pp. 356–362). Turku, Finland: Linköping University Electronic Press.

Korsgaard, K., Vitger, M., & Hannibal, S. (2015). Opdagende skrivning - en vej ind I læsning [Explorative writing - a pathway to reding]. Dansklærerforening.

Kumar, V., & Boulanger, D. (2020). Explainable automated essay scoring: Deep learning really has pedagogical value. *Frontiers in Education*, *5*. https://doi.org/10.3389/feduc.2020.572367.

Lee, A. V. Y., Luco, A. C., & Tan, S. C. (2023). A human-centric automated essay scoring and feedback system for the development of ethical reasoning. *Educational Technology & Society*.

Liberg, C. (1990). *Learning to read and write. Reports from Uppsala university linguistics: Vol. 20*. Department of Linguistics, Uppsala University.

Lorenzen, S., Hjuler, N., & Alstrup, S. (2019). Investigating writing style development in high school. In *Proceedings of the 12th international conference on educational data mining (EDM 2019)*.

Ludwig, S., Mayer, C., Hansen, C., Eilers, K., & Brandt, S. (2021). Automated essay scoring using transformer models. *Psyche*, *3*, 897–915. https://doi.org/10.3390/psych3040056.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge, UK: Cambridge University Press.

Mayfield, E., & Black, A. W. (2020). Should you fine-tune BERT for automated essay scoring? In *Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications* (pp. 151–162). Seattle, WA, USA: Association for Computational Linguistics.

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, *22*, 276–282.

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). *Automated evaluation of text and discourse with coh-metrix*. Cambridge University Press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, arXiv:1301.3781 [abs].

Nivre, J., Zeman, D., Ginter, F., & Tyers, F. (2017). Universal dependencies. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Tutorial abstracts, association for computational linguistics*.

Palmér, A. (2018). Kvantiteter i kvalitativt bedömda elevtexter - framtida verktyg för rättvis bedömning? *Acta Didactica Norge*, *12*, Article 9. https://doi.org/10.5617/adno.6357.

Pedregosa-Izquierdo, F. (2015). *Feature extraction and supervised learning on fMRI: From practice to theory. Theses*. Université Pierre et Marie Curie - Paris VI.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations*.

Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: A systematic literature review. *Artificial Intelligence Review*, *55*, 2495–2527.

Rasch, G. (1960). On general laws and the meaning of measurement in psychology. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability: Held at the statistical laboratory* (p. 321). Univ. of California Press.

Rijsbergen, C. J. V. (1979). *Information retrieval* (2nd ed.). Butterworth-Heinemann.

Saliu-Abdulahi, D., Hellekjær, G. O., & Hertzberg, F. (2017). Teachers' (formative) feedback practices in efl writing classes in Norway.

Stan Development Team (2012). Stan modeling language user's guide and reference manual, version 1.0.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B, Methodological*, *36*, 111–147.

Taghipour, K. (2017). Robust trait-specific essay scoring using neural networks and density estimators. Ph.D. thesis, Singapore: National University of Singapore.

Uto, M., Xie, Y., & Ueno, M. (2020). Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th international conference on computational linguistics, international committee on computational linguistics* (pp. 6077–6088).

Wang, Y., Wang, C., Li, R., & Lin, H. (2022). On the use of BERT for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: Human language technologies, association for computational linguistics*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Davison, J. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations, association for computational linguistics* (pp. 38–45), Online.