



Introducing a Framework to Assess Newly Created Questions with Natural Language Processing

Luca Benedetto¹ , Andrea Cappelli², Roberto Turrin²,
and Paolo Cremonesi¹ 

¹ Politecnico di Milano, Milan, Italy

{luca.benedetto,paolo.cremonesi}@polimi.it

² Cloud Academy Sagl, Mendrisio, Switzerland

{andrea.cappelli,roberto.turrin}@cloudacademy.com

Abstract. Statistical models such as those derived from Item Response Theory (IRT) enable the assessment of students on a specific subject, which can be useful for several purposes (e.g., learning path customization, drop-out prediction). However, the questions have to be assessed as well and, although it is possible to estimate with IRT the characteristics of questions that have already been answered by several students, this technique cannot be used on newly generated questions. In this paper, we propose a framework to train and evaluate models for estimating the difficulty and discrimination of newly created Multiple Choice Questions by extracting meaningful features from the text of the question and of the possible choices. We implement one model using this framework and test it on a real-world dataset provided by *CloudAcademy*, showing that it outperforms previously proposed models, reducing by 6.7% the RMSE for difficulty estimation and by 10.8% the RMSE for discrimination estimation. We also present the results of an ablation study performed to support our features choice and to show the effects of different characteristics of the questions' text on difficulty and discrimination.

Keywords: Natural language processing · Item Response Theory · Learning analytics

1 Introduction

Modeling the skill level of students and how it evolves over time is known as Knowledge Tracing (KT), and it can be leveraged to improve the learning experience, for instance suggesting tailored learning content or detecting students in need of further support. KT is most commonly performed with logistic models or neural networks. Although neural models often reach the best accuracy in predicting the correctness of students' answers, they do not provide easy explanations of their predictions. Logistic models such as Item Response Theory (IRT), instead, estimate latent traits of students and questions (e.g., numerical

values representing skill level and difficulty level) and use those to predict future answers. IRT leverages the answers given by a student to a set of calibrated questions (i.e., whose latent traits are known) to estimate her skill level, by finding the skill value that maximizes the likelihood of the observed results. Questions' latent traits are non-observable parameters which have to be estimated and, if such estimation is not accurate, it affects the students' assessment and impacts the overall efficacy of the system (e.g., suggesting wrongly targeted learning content). Also, an accurate calibration of the questions allows to identify the ones that are not suited for scoring students because they cannot discriminate between different skill levels. For instance, questions that are too difficult or too easy are answered in the same way by all the students, and questions that are unclear (e.g., due to poor wording) are answered correctly or wrongly independently of the knowledge of the students. Questions' latent traits are usually estimated with one of two techniques: they are either i) hand-picked by human experts or ii) estimated with pretesting. Both approaches are far from optimal: manual labeling is intrinsically subjective, thus affected by high uncertainty and inconsistency; pretesting leads to a reliable and fairly consistent calibration but introduces a long delay before using new questions for scoring students [29].

Recent works tried to overcome the problem of calibrating newly-generated questions by proposing models capable of estimating their characteristics from the text: with this approach, it is possible to immediately obtain an estimation of questions' latent traits and, if necessary, this initial estimation can be later fine-tuned using students' answers. However, most works targeted either the *wrongness* or the *p-value* of each question (i.e., the fraction of wrong and correct answers, respectively), which are approximations of the actual difficulty; [4] focus on latent traits as defined in IRT (i.e., difficulty and discrimination). This work introduces *text2props*, a framework to train and evaluate models for calibrating newly created Multiple-Choice Questions (MCQ) from the text of the questions and of the possible choices. The framework is made of three modules for i) estimating ground truth latent traits, ii) extracting meaningful features from the text, and iii) estimating question's properties from such features. The three modules can be implemented with different components, thus enabling the usage of different techniques at each step; it is also possible to use predefined ground truth latent traits, if available. We show the details of a sample model implemented with *text2props* and present the results of experiments performed on a dataset provided by the e-learning provider *CloudAcademy*¹. Our experiments show an improvement in the estimation of both difficulty and discrimination: specifically, reaching a 6.7% reduction in the RMSE for difficulty estimation (from 0.807 to 0.753) and 10.8% reduction in the RMSE for discrimination estimation (from 0.414 to 0.369). We also present an ablation study to empirically support our choice of features, and the results of an experiment on the prediction of students' answers, to validate the model using an observable ground truth. The contributions of this work are: i) the introduction of *text2props*, a framework to implement models for calibrating newly created MCQ, ii) the implementation of

¹ <https://cloudacademy.com/>.

a sample model that outperforms previously proposed models, iii) an ablation study to support our choice of features in the sample model, iv) publication of the framework’s code to foster further research². This document is organized as follows: Sect. 2 presents the related works, Sect. 3 introduces *text2props*, Sect. 4 describes the dataset and the sample model, Sect. 5 presents the results of the experiments, Sect. 6 concludes the paper.

2 Related Work

2.1 Students’ Assessment

Knowledge Tracing (KT) was pioneered by Atkinson [3] and, as reported in a recent survey [2], is most commonly performed with logistic models (e.g., IRT [27], Elo rating system [25]) or neural networks [1, 22]. Recent works on students’ performance prediction claim that Deep Knowledge Tracing (DKT) (i.e., KT with neural networks [22]) outperforms logistic models in predicting the results of future exams [1, 6, 32, 33], but this advantage is not fully agreed across the community [8, 20, 28, 31]. Also, DKT predictions do not provide an explicit numerical estimation of the skill level of the students or the difficulty of the questions. Recent works [17, 30] attempted to make DKT explainable by integrating concepts analogous to the latent traits used in logistic models, but being much more expensive from a computational point of view and without reaching the same level of explainability as logistic models. Thus, logistic models are usually chosen when interpretable latent traits are needed. In this work, we use Item Response Theory (IRT) [12], that estimates the latent traits of students and questions involved in an exam. We consider the two-parameters model, which associates to each item two scalars: the difficulty and the discrimination. The difficulty represents the skill level required to have a 50% probability of correctly answering the question, while the discrimination determines how rapidly the odds of correct answer increase or decrease with the skill level of the student.

2.2 NLP for Latent Traits Estimation

The idea of inferring properties of a question from its text is not new; however, most of previous works did not focus on difficulty estimation. The first works focused on text readability estimation [9, 16]. In [14] the authors use a neural network to extract from questions’ text the topics that are assessed by each question. Wang et al. in [26] and Liu et al. in [18] proposed models to estimate the difficulty of questions published in community question answering services leveraging the text of the question and some domain specific information which is not available in the educational domain, thus framing the problem differently. Closer to our case are some works that use NLP to estimate the difficulty of assessment items, but most of them measured questions’ difficulty as the fraction of students that answered incorrectly (i.e., the *wrongness*) or correctly (i.e., the

² <https://github.com/lucabenedetto/text2props>.

p-value), which are arguably a more limited estimation than the IRT difficulty, as they do not account for different students’ skill levels. Huang et al. [13] propose a neural model to predict the difficulty of “reading” problems in Standard Tests, in which the answer has to be found in a text provided to the students together with the question. Their neural model uses as input both the text of the question and the text of the document, a major difference from our case. Yaneva et al. in [29] introduce a model to estimate the *p-value* of MCQ from the text of the questions, using features coming from readability measures, word embeddings, and Information Retrieval (IR). In [23] the authors propose a much more complex model, based on a deep neural network, to estimate the *wrongness* of MCQ. In [4] the authors use IR features to estimate the IRT difficulty and the discrimination of MCQ from the text of the questions and of the possible choices. All relevant related works experimented on private datasets and only [4] focuses on IRT latent traits. In this paper, we make a step forward with respect to previous research by introducing *text2props*, a modular framework to train and evaluate models for estimating the difficulty and the discrimination of MCQ from textual information. Then, we implement a sample model with *text2props* and test it on a sub-sample of a private dataset provided by *CloudAcademy*.

3 The Framework

3.1 Data Format

The *text2props* framework interacts with two datasets: i) the *Questions* (Q) dataset contains the textual information, ii) the *Answers* (A) dataset contains the results of the interactions between students and questions. Specifically, Q contains, for each question: i) ID of the question, ii) text of the MCQ, iii) text of all the possible choices, and iv) which are the correct choices and which the distractors. A , instead, contains for each interaction: i) ID of the student, ii) ID of the question, iii) correctness of student’s answer, and iv) timestamp of the interaction. The interactions in A are used to obtain the ground truth latent traits of each question, which are used as target values while training the estimation of latent traits from textual information.

3.2 Architecture

Three modules compose *text2props*: i) an IRT estimation module to obtain ground truth latent traits, ii) a feature engineering module to extract features from text, and iii) a regression module to estimate the latent traits from such features. At training time all the modules are trained, while only the feature engineering module and the regression module are involved in the inference phase.

Figure 1 shows how the three modules interact with the datasets during **training**. A split stratified on the questions is performed on A , producing the dataset for estimating the ground truth latent traits (A_{GTE}) and the dataset for evaluating students’ answers prediction (A_{SAP}). This is done in order to have

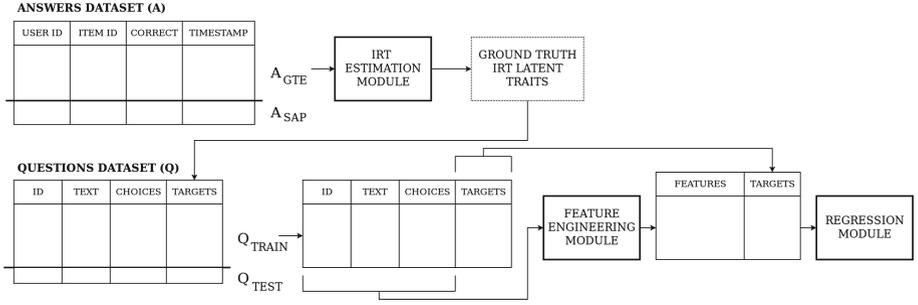


Fig. 1. Framework's architecture and interactions with the datasets during training.

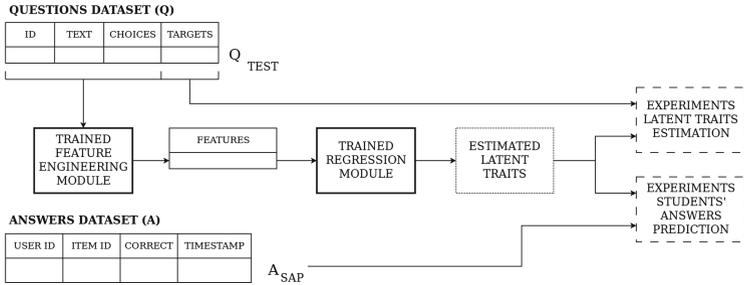


Fig. 2. Framework's architecture and interactions with the datasets during inference.

all the questions in both datasets and, therefore, be able to obtain the ground truth latent traits of all the questions from A_{GTE} and, later, perform the experiments on students' answers prediction using previously unseen interactions. The ground truth latent traits obtained with the IRT estimation module from A_{GTE} are then stored in Q , in order to be used as target values in the regression module. Then, a split is performed on Q , obtaining the dataset used to train the feature engineering and regression modules (Q_{TRAIN}) and the dataset to test them (Q_{TEST}). Lastly, the textual information of Q_{TRAIN} is used by the feature engineering module to extract numerical features, which are then used together with the ground truth latent traits to train the regression module.

During the **inference** phase, pictured in Fig. 2, the trained feature engineering module is fed with the textual information of the questions in Q_{TEST} , and extracts the features that are given to the trained regression module to estimate the latent traits. These estimated latent traits can then be used for evaluating i) latent traits estimation, directly comparing them with the ground truth latent traits (in Q_{TEST}), and ii) students' answers prediction, comparing the predictions with the true answers (in A_{SAP}).

4 Experimental Setup

4.1 Sample Model

In order to implement a model using *text2props*, it is sufficient to define the three modules. In the sample model used for the experiments, the calibration module performs the estimation of the IRT difficulty and discrimination of each question; these two latent traits are then used as ground truth while training the part of the model that performs the estimation from text. The regression module contains two Random Forests to estimate the difficulty and discrimination. The feature engineering module is made of three components to compute: i) readability features, ii) linguistic features, iii) Information Retrieval features.

- **Readability indexes** are measures designed to evaluate how easy a text is to understand, thus they can prove useful for estimating question’s properties, as suggested in [29]. In particular, we use: *Flesch Reading Ease* [10], *Flesch-Kincaid Grade Level* [15], *Automated Readability Index* [24], *Gunning FOG Index* [11], *Coleman-Liau Index* [7], and *SMOG Index* [21]. All these indexes are computed with deterministic formulas from measures such as the number of words and the average word length.
- The usage of **linguistic features** is motivated by [9], in which they proved useful for readability estimation. The following features are used: *Word Count Question*, *Word Count Correct Choice*, *Word Count Wrong Choice*, *Sentence Count Question*, *Sentence Count Correct Choice*, *Sentence Count Wrong Choice*, *Average Word Length Question*, *Question Length divided by Correct Choice Length*, *Question Length divided by Wrong Choice Length*.
- The choice of **Information Retrieval** (IR) features is supported by previous research [4] and by the idea that there must be a relation between the latent traits of a MCQ and the words that appear in the text. We i) preprocess the texts using standard steps of NLP [19], ii) consider both the text of the question and the text of the possible choices by concatenating them, and iii) use features based on Term Frequency-Inverse Document Frequency (TF-IDF). However, instead of keeping only the words whose frequency is above a certain threshold (as in [4]), we define two thresholds - tuned with cross-validation - to remove i) corpus-specific stop words (i.e., words with frequency above SUP) and ii) very uncommon words (i.e., with frequency below INF).

4.2 Experimental Dataset

All previous works experimented on private data collections [4, 13, 23, 29] and, similarly, we evaluate this framework on a private dataset, which is a sub-sample of real world data coming from the e-learning provider *CloudAcademy*. Dataset *Q* contains about 11 K multiple-choice questions and they have 4 possible answers; for some questions, there is more than one correct answer and, in that case, the student is asked to select all the correct choices. Dataset *A*, which is used for estimating the ground truth latent traits and for the experiments on students’

answers prediction, contains about 2M answers. Also, it is filtered in order to keep only the students and the questions that appear in at least 100 different interactions; thus we assume that the IRT-estimated latent traits are accurate enough to be used as ground truth for this study.

5 Results

5.1 Latent Traits Estimation

The sample model used for the comparison with the state of the art was chosen from a pool of models, all implemented with *text2props*. All these models had the same IRT estimator module and the same feature engineering module, containing the three components described in Sect. 4.1, but they were implemented with different algorithms in the regression module: specifically, we tested Random Forests (RF), Decision Trees (DT), Support Vector Regression (SVR), and Linear Regression (LR). For each model, hyperparameter tuning was performed via 10-fold randomized cross-validation [5]. The results of this preliminary experiments for choosing the sample model are displayed in Table 1, presenting for each candidate model the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) for difficulty estimation and discrimination estimation, separately on a validation set held-out from the test set and on the remaining test set. The two errors measure how accurate the sample model is by comparing the latent traits (i.e., difficulty and discrimination) estimated from text with the ground truth values obtained with IRT estimation. As baseline, we consider a majority prediction, which assigns to all the questions the same difficulty and discrimination, obtained by averaging the training latent traits. All the models outperform the majority baseline, and the RF leads to the best performance in both cases; thus, that is the model which will be used as sample model for the rest of the experiments and the comparison with the state of the art.

Table 1. Preliminary experiments for choosing the sample model.

Regression module	Difficulty estimation				Discrimination estimation			
	Validation set		Test set		Validation set		Test set	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
RF	0.739	0.575	0.753	0.587	0.393	0.296	0.369	0.287
DT	0.748	0.586	0.826	0.636	0.393	0.295	0.375	0.290
SVR	0.797	0.632	0.804	0.629	0.394	0.298	0.379	0.296
LR	0.752	0.599	0.779	0.607	0.397	0.298	0.378	0.293
Majority	–	–	0.820	0.650	–	–	0.502	0.427

Table 2. Comparison with state of the art.

Model	Difficulty estimation			Discrimination estimation		
	Range	RMSE	Relative RMSE	Range	RMSE	Relative RMSE
Our model	[-5; 5]	0.753	7.53%	[-1; 2.5]	0.369	9.22%
R2DE [4]	[-5; 5]	0.807	8.07%	[-1; 2.5]	0.414	10.35%
Qiu et al. [23]	[0; 1]	0.1521	15.21%	-	-	-
Huang et al. [13]	[0; 1]	0.21	21%	-	-	-
Yaneva et al. [29]	[0; 100]	22.45	22.45%	-	-	-

Table 2 compares the model implemented with *text2props* with the state of the art for difficulty and discrimination estimation. Considering difficulty estimation, our model reduces the RMSE by 6.7% (from 0.807 to 0.753) with respect to R2DE, which was implemented using the code publicly available³, re-trained and tested on the new dataset. The other works experimented on private datasets and could not be directly re-implemented on our dataset, therefore a comparison on the same dataset was not straightforward; however, as suggested in [4], we can still gain some insight by performing a comparison on the Relative RMSE, which is defined as: $\text{RMSE}/(\text{difficulty}_{\max} - \text{difficulty}_{\min})$. The Relative RMSE of the sample model is smaller than the ones obtained in previous research and, although this does not guarantee that it would perform better than the others on every dataset, it suggests that it might perform well. The part of the table about discrimination estimation contains only two lines since this and R2DE are the only works that estimate both the difficulty and the discrimination. Again, our model outperforms R2DE, reducing the RMSE from 0.414 to 0.369.

5.2 Students' Answers Prediction

The accuracy of latent traits estimation is commonly evaluated by measuring the error with respect to ground truth latent traits estimated with IRT. However, although IRT is a well-established technique, such latent traits are non observable properties, and we want to validate our model on an observable ground truth as well, therefore we evaluate the effects that it has in predicting the correctness of students' answers. Students' Answers Prediction (SAP) provides an insight on the accuracy of latent traits estimation because questions' latent traits are a key element in predicting the correctness of future answers. Indeed, given a student i with estimated skill level $\hat{\theta}_i$ and a question j with difficulty b_j and discrimination a_j , the probability of correct answer is computed as

$$P_C = \frac{1}{1 + e^{-1.7a_j \cdot (\hat{\theta}_i - b_j)}} \quad (1)$$

³ <https://github.com/lucabenedetto/r2de-nlp-to-estimating-irt-parameters>.

The skill level $\tilde{\theta}_i$ is estimated from the answers previously given by the student:

$$\tilde{\theta}_i = \max_{\theta} \left[\prod_{q_j \in Q_C} \frac{1}{1 + e^{-1.7a_j \cdot (\theta - b_j)}} \cdot \prod_{q_j \in Q_W} \left(1 - \frac{1}{1 + e^{-1.7a_j \cdot (\theta - b_j)}} \right) \right] \quad (2)$$

where Q_C and Q_W are sets containing the questions correctly and wrongly answered by the student, respectively.

Known the ordered sequence of interactions, SAP is performed as follows:

1. given the latent traits of a question (b_j , a_j) and the student’s estimated skill level ($\tilde{\theta}_i$, possibly unknown), the probability of correct answer is computed;
2. if the probability is greater than 0.5 we predict a correct answer;
3. the real answer is observed and compared to the prediction (this is the comparison used to compute the evaluation metrics);
4. the real answer is used to update the estimation of the student’s skill level;
5. these steps are repeated for all the items the student interacted with.

By using in the two equations above latent traits coming from different sources, we compare the accuracy of SAP obtained i) with the latent traits estimated with our model, and ii) with ground truth IRT latent traits. Table 3 displays the results of the experiment, showing also as baseline a simple majority prediction. As metrics, we use Area Under Curve (AUC), accuracy, precision and recall on correct answers, and precision and recall on wrong answers. The table shows that our model performs consistently better than the majority baseline and fairly closely to IRT - which is an upper threshold - suggesting that the estimation of latent traits from text can be successfully used as initial calibration of newly generated items. However, it might still be convenient to fine-tune such estimation when the data coming from student interactions becomes available.

Table 3. Students’ answers prediction.

Model	AUC	Accuracy	Correct		Wrong	
			Precision	Recall	Precision	Recall
IRT	0.74	0.683	0.744	0.735	0.589	0.599
Our model	0.66	0.630	0.707	0.678	0.521	0.555
Majority	0.50	0.613	0.613	1.0	–	0.000

5.3 Ablation Study

The objective of this ablation study is to i) empirically support our choice of features and ii) assess the impact of specific features on the estimation. Table 4 presents the RMSE and the MAE for difficulty estimation and discrimination

estimation. In all cases, we use Random Forests in the regression module, since it seemed to be the most accurate and robust approach, according to the preliminary experiments; as baseline, we consider the majority prediction. The combination of all the features leads to the smallest errors, thus suggesting that all the features bring useful information. The IR features seem to provide the most information when considered alone: this is reasonable, since they have two parameters that can be tuned to improve the performance. The smallest error is usually obtained when some terms are removed from the input text; most likely, both corpus specific stop-words and terms which are too rare only introduce noise. It is interesting to notice that readability and linguistic features seem to be more useful for discrimination than difficulty estimation since, when used alone, they perform similarly to the best performing features.

Table 4. Ablation study.

Features	Difficulty estimation				Discrimination estimation			
	INF	SUP	RMSE	MAE	INF	SUP	RMSE	MAE
IR + Ling. + Read.	0.02	0.92	0.753	0.587	0.02	0.96	0.369	0.287
IR + Ling.	0.02	0.90	0.754	0.587	0.02	0.98	0.370	0.287
IR + Read.	0.02	0.94	0.766	0.597	0.02	0.98	0.370	0.288
IR	0.00	0.92	0.758	0.587	0.02	0.96	0.372	0.289
Read + Ling	–	–	0.791	0.618	–	–	0.373	0.291
Readability	–	–	0.794	0.619	–	–	0.374	0.292
Linguistic	–	–	0.791	0.620	–	–	0.375	0.292
Majority	–	–	0.820	0.650	–	–	0.502	0.427

6 Conclusions

In this paper we introduced *text2props*, a framework that allows the training and evaluation of models for calibrating newly created Multiple-Choice Questions from textual information. We evaluated a sample model implemented with *text2props* on the tasks of latent traits estimation and students’ answers prediction, showing that models implemented with this framework are capable of providing an accurate estimation of the latent traits, thus offering an initial calibration of newly generated questions, which can be fine-tuned when student interactions become available. Our model outperformed the baselines reaching a 6.7% reduction in the RMSE for difficulty estimation and 10.8% reduction in the RMSE for discrimination estimation. As for students’ answers prediction, it improved the AUC by 0.16 over the majority baseline, and performed fairly close to the prediction made with IRT latent traits (which is an upper threshold), having an AUC 0.08 lower. Lastly, the ablation study showed that

all features are useful for improving the estimation of the latent traits from text, as the best results are obtained when combining all of them. Future works will focus on exploring the effects of other features on the estimation of latent traits (e.g., word embeddings, latent semantic analysis) and testing the capabilities of this framework to estimate other question's properties. Also, future work should focus on the main limitation of *text2props*, consisting in the fact that it forces the implemented models to have the three-modules architecture presented here; in this case the model implemented with this framework proved effective, but it is not guaranteed that it would work similarly well in other situations.

References

1. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks (2019)
2. Abyaa, A., Idrissi, M.K., Bennani, S.: Learner modelling: systematic review of the literature from the last 5 years. *Educ. Technol. Res. Dev.* **67**, 1–39 (2019)
3. Atkinson, R.C.: Ingredients for a theory of instruction. *Am. Psychol.* **27**(10), 921 (1972)
4. Benedetto, L., Cappelli, A., Turrin, R., Cremonesi, P.: R2DE: a NLP approach to estimating IRT parameters of newly generated questions. In: *Proceedings of the Tenth International Conference on Learning Analytics and Knowledge*, pp. 412–421 (2020)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(Feb), 281–305 (2012)
6. Chen, P., Lu, Y., Zheng, V.W., Pian, Y.: Prerequisite-driven deep knowledge tracing. In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 39–48. IEEE (2018)
7. Coleman, E.B.: On understanding prose: some determiners of its complexity. NSF final report GB-2604 (1965)
8. Ding, X., Larson, E.: Why deep knowledge tracing has less depth than anticipated. In: *The 12th International Conference on Educational Data Mining* (2019)
9. DuBay, W.H.: The principles of readability. Online Submission (2004)
10. Flesch, R.: A new readability yardstick. *J. Appl. Psychol.* **32**(3), 221 (1948)
11. Gunning, R.: *Technique of clear writing* (1968)
12. Hambleton, R.K., Swaminathan, H., Rogers, H.J.: *Fundamentals of Item Response Theory*. Sage, Thousand Oaks (1991)
13. Huang, Z., et al.: Question difficulty prediction for reading problems in standard tests. In: *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
14. Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G., et al.: EKT: exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* **PP**, 1 (2019)
15. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel (1975)
16. Kintsch, W., Vipond, D.: Reading comprehension and readability in educational practice and psychological theory. *Perspect. Learn. Mem.* 329–365 (2014)
17. Lee, J., Yeung, D.Y.: Knowledge query network for knowledge tracing: how knowledge interacts with skills. In: *Proceedings of the 9th International Conference on Learning Analytics and Knowledge*, pp. 491–500. ACM (2019)

18. Liu, J., Wang, Q., Lin, C.Y., Hon, H.W.: Question difficulty estimation in community question answering services. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 85–90 (2013)
19. Manning, C.D., Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT press, Cambridge (1999)
20. Mao, Y., Lin, C., Chi, M.: Deep learning vs. bayesian knowledge tracing: student models for interventions. *JEDM J. Educ. Data Min.* **10**(2), 28–54 (2018)
21. Mc Laughlin, G.H.: Smog grading-a new readability formula. *J. Reading* **12**(8), 639–646 (1969)
22. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, pp. 505–513 (2015)
23. Qiu, Z., Wu, X., Fan, W.: Question difficulty prediction for multiple choice problems in medical exams. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 139–148. ACM (2019)
24. Senter, R., Smith, E.A.: Automated readability index. Technical Report, Cincinnati University, OH (1967)
25. Verhagen, J., Hatfield, D., Arena, D.: Toward a scalable learning analytics solution. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds.) International Conference on Artificial Intelligence in Education, vol. 11626, pp. 404–408. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23207-8_74
26. Wang, Q., Liu, J., Wang, B., Guo, L.: A regularized competition model for question difficulty estimation in community question answering services. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1115–1126 (2014)
27. Wang, X., Berger, J.O., Burdick, D.S., et al.: Bayesian analysis of dynamic item response models in educational testing. *Ann. Appl. Stat.* **7**(1), 126–153 (2013)
28. Wilson, K.H., Karklin, Y., Han, B., Ekanadham, C.: Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation (2016)
29. Yaneva, V., Baldwin, P., Mee, J., et al.: Predicting the difficulty of multiple choice questions in a high-stakes medical exam. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 11–20 (2019)
30. Yeung, C.K.: Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. arXiv preprint [arXiv:1904.11738](https://arxiv.org/abs/1904.11738) (2019)
31. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the Fifth Annual ACM Conference on Learning at Scale, p. 5. ACM (2018)
32. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774. International World Wide Web Conferences Steering Committee (2017)
33. Zhang, L., Xiong, X., Zhao, S., Botelho, A., Heffernan, N.T.: Incorporating rich features into deep knowledge tracing. In: Proceedings of the Fourth ACM Conference on Learning@ Scale (2017), pp. 169–172. ACM (2017)