ScoreCLIQ: A Dynamic LLM-Based Approach for Item Difficulty Estimation

No Author Given

No Institute Given

Abstract. Accurately estimating the difficulty of Multiple-Choice Questions (MCQs) is crucial for creating fair and well-balanced assessments. Traditional methods rely on pretesting, which is expensive and time-consuming. Recent AI-based approaches use deep learning models to predict difficulty scores from MCQ text, but they treat difficulty estimation as a static regression task, failing to account for how small changes in question wording impact difficulty. We introduce ScoreCLIQ, a dynamic difficulty estimator that integrates Large Language Models (LLMs) and Reinforcement Learning (RL) to refine difficulty predictions. ScoreCLIQ follows a three-stage process: (i) training a BERT-based model for baseline difficulty estimation, (ii) optimizing an LLM with RL to paraphrase questions based on difficulty labels, and (iii) refining the estimator using regularized loss on the LLM-generated paraphrases. Experiments show that ScoreCLIQ outperforms existing models, reducing RMSE by ∼17% compared to the best state-of-the-art difficulty estimation method. This demonstrates the effectiveness of combining LLMs and reinforcement learning for improved MCQ difficulty estimation.

Keywords: Item Difficulty Estimation · LLM · Reinforcement Learning

1 Introduction

Multiple-Choice Questions (MCQs) are widely used in standardized tests to assess knowledge across various fields, including education, medicine, and professional certification exams [1,2]. A key challenge in designing these assessments is ensuring that each question has an appropriate difficulty level. Accurate difficulty estimation helps maintain a balanced mix of easy, moderate, and hard questions, improving test validity and adaptive learning systems [1]. Traditionally, estimating MCQ difficulty relies on *pretesting*, where new questions are added to live exams to analyze candidate responses [3]. However, this approach is costly, slow, and impractical for large-scale assessments.

With advances in Natural Language Processing (NLP), automated difficulty estimation has gained attention. Early methods relied on handcrafted linguistic features and statistical models, while recent approaches use deep learning, particularly transformer-based models, to analyze question text [4]. However, these models typically treat difficulty estimation as a fixed regression task, predicting scores without considering how small wording changes can significantly impact

difficulty [1]. In reality, minor modifications in phrasing, structure, or distractor choices can alter how challenging a question appears to students.

To address these limitations, we propose ScoreCLIQ, a dynamic difficulty estimator that integrates Large Language Models (LLMs) and Reinforcement Learning (RL) to refine difficulty predictions. ScoreCLIQ follows a three-stage process: (a) Baseline Estimation - A *BERT-based* model is fine-tuned to predict difficulty from MCQ text. (b) LLM-Guide Refinement – A guide *LLM* enhanced using *reinforcement learning* to paraphrase questions for better alignment with their true difficulty scores and (c) Estimator Improvement – The refined paraphrases generated from the *reinforcement learning-enhanced LLM* are used to further fine-tune the difficulty estimator with a regularized loss function, improving accuracy.

By incorporating dynamic question modification, ScoreCLIQ adapts to variations in question phrasing that affect difficulty. Our experiments demonstrate that ScoreCLIQ outperforms existing difficulty estimation methods, providing a more accurate and adaptable solution for automated MCQ difficulty prediction.

2 Related Works

Estimating the difficulty of multiple-choice questions (MCQs) has evolved from traditional statistical methods to advanced machine learning techniques. Early approaches relied on Item Response Theory (IRT) [5], which models the probability of a correct response based on both the item's difficulty and the examinee's ability. While effective, IRT requires extensive response data, making it less practical for newly developed items. To address this, researchers have explored linguistic features to predict item difficulty. Traditional machine learning models utilized handcrafted features such as word complexity and sentence length [6]. However, these models often struggled with domain-specific nuances, especially in fields like medical education.

The advent of deep learning and transformer-based models, such as BERT [4], has enhanced the ability to capture semantic nuances in questions. For instance, Tack et al. [7] demonstrate that simpler models like Lasso and Random Forest, when combined with linguistic and clinical embeddings, can outperform more complex models in predicting item difficulty and response time. Similarly, Gombert et al. [8] employed scalar-mixed transformer encoder models with rational network regression heads, achieving notable success in the BEA 2024 Shared Task. Despite these advancements, many models treat difficulty estimation as a static task, not accounting for how subtle changes in question phrasing can impact difficulty. Our approach, ScoreCLIQ, addresses this by integrating Large Language Models (LLMs) and reinforcement learning to dynamically refine difficulty predictions, ensuring that MCQs align more closely with their intended difficulty levels.

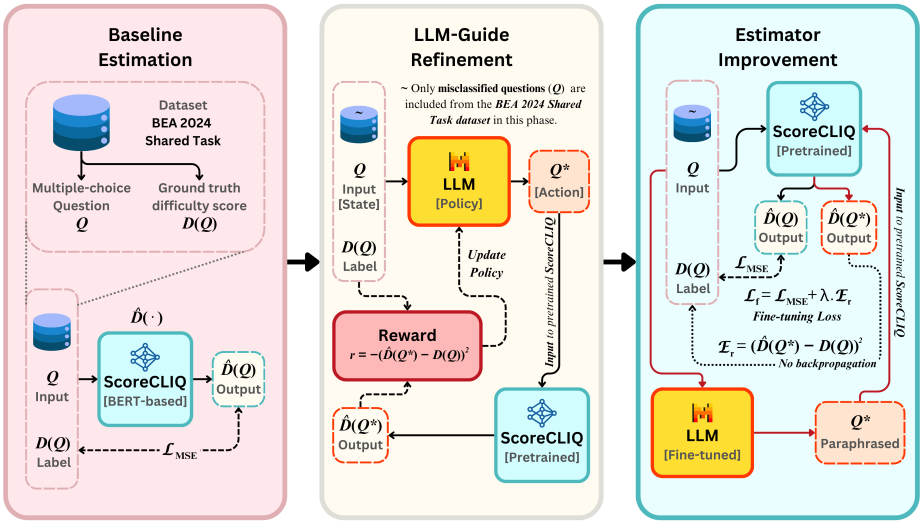

Fig. 1. The detailed three-stage training pipeline of ScoreCLIQ.

3 Methodology

Most frameworks, though harnessing the power of LLMs, are treating difficulty estimation like a static regression task [9,10,11,12,13], even when it is popular that LLM itself undergoes RLHF[1] for dynamically aligning itself with human feedback in its standard training pipeline [14,15,16]. Inspired from the idealogy of the standard LLM training pipeline, we introduce ScoreCLIQ: a BERT-based item difficulty estimator for clinical MCQs with a three-stage training pipeline. The proposed pipeline effectively improves item difficulty estimation using LLM as a guide. The stages are: i) Baseline Estimation: fine-tuning a BERT-based [4] difficulty estimator to serve as the core of ScoreCLIQ; ii) LLM-Guide Refinement: fine-tuning the LLM as a paraphraser via Reinforcement Learning with rewards based on pre-trained ScoreCLIQ; iii) Estimator Improvement: fine-tuning and improving the ScoreCLIQ estimator based on a regularized objective function. Fig. 1 illustrates the details of the proposed pipeline, which we discuss next.

Baseline Estimation: We use pre-trained BERT as the baseline ScoreCLIQ estimator for its simplicity, flexibility, and resource efficiency [4]. We add a fully connected layer to the end of pre-trained BERT and fine-tune it to predict the difficulty score. We begin the fine-tuning process of the ScoreCLIQ model $\hat{D}(\cdot)$, using multiple-choice questions $\{Q_i\}_{i=1}^{N}$ and their corresponding ground-truth difficulty scores $\{D(Q_i)\}_{i=1}^{N}$ from our preprocessed dataset $\boldsymbol{\mathcal{D}}$. The loss objective used for initializing the ScoreCLIQ $\hat{D}(\cdot)$ is simply the mean squared error loss

[1] RLHF: Reinforcement Learning with Human Feedback

between predictions $\hat{D}(Q_i)$ and labels $D(Q_i)$:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{D}(Q_i) - D(Q_i))^2 \tag{1}$$

where, N is number of samples in the training data. After initial fine-tuning, we evaluate ScoreCLIQ across the train and validation set and compute the error per question (ϵ_i) using the equation, $\epsilon_i = |\hat{D}(Q_i) - D(Q_i)|$, to filter out the significantly misclassified questions ($\tilde{\boldsymbol{\mathcal{D}}} \subset \boldsymbol{\mathcal{D}}$) based on a threshold t. We use $\tilde{\boldsymbol{\mathcal{D}}}$ for the subsequent stages.

LLM-Guide Refinement: We now move on to the second stage of our pipeline where we fine-tune an LLM to dynamically improve the ScoreCLIQ. The primary role of LLM is to paraphrase the misclassified question Q_i to its refined version Q_i^* so that it aligns better with the ground-truth difficulty score $D(Q_i)$. We can simply define this as a dynamic modification function $f(\cdot)$:

$$Q_i^* = f(Q_i, \hat{D}(Q_i), D(Q_i); \theta) \tag{2}$$

where, $f(\cdot)$ is an LLM-based correction function which paraphrases based on $\hat{D}(Q_i)$, $D(Q_i)$ and parameters θ. To prompt the LLM appropriately to use $\hat{D}(Q_i)$ and $D(Q_i)$, we use a Markov Decision Process (MDP) based reinforcement learning (RL) approach [17]. MDP assists the LLM to be a better paraphraser aligning Q_i^* with human-annotated difficulty scores. We train the $f(\cdot)$ as the policy, having an initial state Q_i, generating Q_i^* as an action and getting reward $r = -(\hat{D}(Q_i) - D(Q_i))^2$ for completion of every action instance. The RL model updates the policy model using the Bellman update rule with a discount rate of 1 ($\gamma = 1$). After successfully fine-tuning the LLM using RL, it is now ready to be used as a guide to further fine-tune ScoreCLIQ.

Estimator Improvement: Finally, in the third stage, we fine-tune ScoreCLIQ on the misclassified samples using the LLM from previous stage. The fine-tuning procedure is very similar to the pre-training approach, the only change being the addition of a regularization term in the final loss. For each Q_i in the misclassified dataset, we generate a Q_i^* from the fine-tuned LLM. Then we predict the difficulty score $\hat{D}(Q_i^*)$ for each Q_i^*. Now using the resulting $\hat{D}(Q_i^*)$ and the true $D(Q_i)$ we devised a regularized loss to fine-tune ScoreCLIQ, as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{MSE} + \lambda(\hat{D}(Q_i^*) - D(Q_i))^2 \tag{3}$$

where, λ is a hyperparameter to control the weight of the regularization term in the loss objective. Note that the Q_i^* and Q_i are not treated as separate training samples, instead, difficulty scores $D(Q_i)$ and $\hat{D}(Q_i^*)$ are used simultaneously in the loss (see Eqns. 1, 3). After finishing all the training phases, the ScoreCLIQ updated in this stage is expected to perform better on item difficulty estimation than the baseline (trained in stage 1).

4 Experimental Details

As mentioned in the previous section, we have used the item difficulty task related dataset from BEA 2024 shared task [1] for training ScoreCLIQ. Fig. 2

Format of an MCQ - Q_i

```
{ItemStem_Text}
A. {Answer__A}
B. {Answer__B}
C. {Answer__C}
D. {Answer__D}
E. {Answer__E}
⋮
J. {Answer__J}

Correct Answer: {Answer_Key}. {Answer_Text}
```

Sample of an MCQ - Q_i

A 45-year-old man is brought to the emergency department because of moderate chest pain after a generalized tonic-clonic seizure 30 minutes ago. He has seizure disorder for which he has taken carbamazepine and phenobarbital for the past 20 years. X-ray of the chest shows generalized osteopenia with several rib fractures. Which of the following is the most likely nutritional deficiency?
A. Folic acid
B. Magnesium
C. Vitamin B2 (riboflavin)
D. Vitamin D

Correct Answer: D. Vitamin D

Fig. 2. Left: Format of each MCQ; and Right: a sample MCQ (Q_i) from the dataset.

left shows the format of the questions in the dataset. For each sample, we only extract the question along with the options, correct answer key, and correct answer text - combining all of these into the Q_i as shown in Fig. 2 right. We further extract ground-truth difficulty scores $D(Q_i)$ for each question Q_i to create the processed dataset $\boldsymbol{\mathcal{D}}$.

We split the training data containing 466 questions from BEA'24 shared task [1] into train:validation ratio of 70:30. The test set containing 201 questions is kept untouched for maintaining fairness during model evaluation against the state-of-the-art benchmarks. As previously mentioned, the ScoreCLIQ model is initialized as a BERT model [4] finetuned on the item difficulty dataset. To further improve ScoreCLIQ, we use Mistral-7B-Instruct-v0.3 [18] for all the experiments. In the next section, we present the results of ScoreCLIQ and compare them with the recent benchmarks outlined in [1]. We use $\lambda = 0.99$, $t = 0.04$ for final experiments. We also benchmark the effect of hyperparameters.

Table 1. Results of Benchmarking ScoreCLIQ's performance against state-of-the-art models. Run and RMSE in the last two columns are shown in the form of A(B), where A and B are team/model's best and baseline runs/results respectively.

Team/Model Name	**Run**	**RMSE** ↓
ScoreCLIQ	**stage 3** (stage 1 baseline)	**0.246** (0.327)
EduTec	electra (roberta)	0.299 (0.304)
UPN-ICC	run1	0.303
ITEC	RandomForest (Ensemble)	0.305 (0.308)
BC	ENSEMBLE (FEAT, ROBERTA)	0.305 (0.305, 0.306)
Scalar	Predictions	0.305
UnibucLLM	run1	0.308
EDU	Run3 (Run1)	0.308 (0.308)
UNED	run3	0.308
Rishikesh	1	0.310
Iran-Canada	run2	0.311
Dummy Regressor	baseline	0.311

5 Results

The comparative analysis of ScoreCLIQ and state-of-the-art (SOTA) based on RMSE is shown in Table 1. Results in row 1 reveal that the aligned LLM is successful in improving the performance of ScoreCLIQ (stage 3 of Sec 3) with an RMSE of 0.246 compared to its own baseline (stage 1) having an RMSE of 0.327. Refined ScoreCLIQ also performs better than other SOTA models, in terms of RMSE. Also, the significant improvement of RMSE by ∼17% w.r.t. EduTec (see first two rows of Table 1), compared to ∼4% improvement of EduTec w.r.t. Dummy Regressor (2nd and last rows of of Table 1), shows the superiority of our methodology against SOTA difficulty score estimation approaches. It is also important to note that ScoreCLIQ improves maximum compared to its baseline, in contrast to the gaps in other methods and their baselines (in parenthesis), as outlined in Table 1. This highlights the effect of the LLM-guide refinement and dynamic estimation in our pipeline.

Table 2. Benchmarking the Effect of Hyperparameters in improving ScoreCLIQ.

Model Config	**Threshold (t)**	**RMSE ↓**
ScoreCLIQ ($\lambda = 0.99$)	0.1	0.291
	0.08	0.282
	0.06	0.273
	0.04	**0.246**
	0.02	0.251

Model Config	λ	**RMSE ↓**
ScoreCLIQ ($t = 0.04$)	1	0.249
	0.99	**0.246**
	0.98	**0.246**
	0.97	0.247
	0.96	0.250

We show the effect of hyperparameters on ScoreCLIQ in Table 2. Keeping the $\lambda = 0.99$ constant, reducing the threshold t from 0.1 to 0.04 increases performance by raising the complex examples for refinement. Reducing t further introduces noisy samples in the refinement stage and affects the performance. Keeping the $t = 0.04$ constant, reducing the $\lambda = 0.99$ from 1 to 0.98 improves performance balancing the two terms in the loss Eq. 3 well. Further reduction in values of λ reduces the effect of LLM-guide refinement (second part of Eq. 3 loss) in the dynamic updates. The studies validate the robustness of ScoreCLIQ's methodology, showing that careful selection of hyperparameters ($\lambda = 0.99, t = 0.04$) leads to optimal performance. The model maintains its effectiveness across different configurations, highlighting its stability and reliability in real-world applications.

6 Conclusion

We presented ScoreCLIQ, a dynamic difficulty estimator guided by a reinforcement learning (RL) enhanced Large Language Model (LLM). The results demonstrate that ScoreCLIQ improves over its static baseline and state-of-the-art (SOTA) models in item difficulty score estimation, achieving the lowest RMSE of 0.246. This marks a significant ∼17% improvement compared to other competing methods. The integration of an LLM-guide refinement and dynamic estimation has been particularly effective in enhancing ScoreCLIQ's predictive capabilities. Overall, ScoreCLIQ has set a new benchmark in the field of item difficulty estimation for clinical MCQs.

References

1. Yaneva, V., North, K., Baldwin, P., Ha, L.A., Rezayi, S., Zhou, Y., Ray Choudhury, S., Harik, P., Clauser, B.: Findings from the first shared task on automated prediction of difficulty and response time for multiple-choice questions. In: Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024). pp. 470–482. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024), https://aclanthology.org/2024.bea-1.39/
2. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683 (2017)
3. Settles, B., LaFlair, G.T., Hagiwara, M.: Machine learning–driven language assessment. Transactions of the Association for Computational Linguistics **8**, 247–263 (Dec 2020). https://doi.org/10.1162/tacl_a_00310, http://dx.doi.org/10.1162/tacl_a_00310
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423/
5. Lord, F.: A Theory of Test Scores. Psychometric Society (1952), https://books.google.co.in/books?id=fxKQ9-FrjgwC
6. Beinborn, L., Zesch, T., Gurevych, I.: Predicting the difficulty of language proficiency tests. Transactions of the Association for Computational Linguistics **2**, 517–530 (2014), https://api.semanticscholar.org/CorpusID:13822684
7. Tack, A., Buseyne, S., Chen, C., D'hondt, R., De Vrindt, M., Gharahighehi, A., Metwaly, S., Nakano, F.K., Noreillie, A.S.: Itec at bea 2024 shared task: Predicting difficulty and response time of medical exam questions with statistical, machine learning, and language models. In: Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024). pp. 512–521 (2024)
8. Gombert, S., Menzel, L., Di Mitri, D., Drachsler, H.: Predicting item difficulty and item response time with scalar-mixed transformer encoder models and rational network regression heads. In: Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024) (2024)
9. Rogoz, A.C., Ionescu, R.T.: UnibucLLM: Harnessing LLMs for automated prediction of item difficulty and response time for multiple-choice questions. In: Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024). pp. 493–502. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024), https://aclanthology.org/2024.bea-1.41/
10. Veeramani, H., Thapa, S., Shankar, N.B., Alwan, A.: Large language model-based pipeline for item difficulty and response time estimation for educational assessments. In: Workshop on Innovative Use of NLP for Building Educational Applications (2024), https://api.semanticscholar.org/CorpusID:270766172
11. Ram, G.V.R., Kesanam, A., M, A.K.: Leveraging physical and semantic features of text item for difficulty and response time prediction of usmle questions. In: Workshop on Innovative Use of NLP for Building Educational Applications (2024), https://api.semanticscholar.org/CorpusID:270766188
12. Dueñas, G., Jimenez, S., Ferro, G.M.: Upn-icc at bea 2024 shared task: Leveraging llms for multiple-choice questions difficulty prediction. In: Workshop on In-

novative Use of NLP for Building Educational Applications (2024), https://api.semanticscholar.org/CorpusID:270766389
13. Gombert, S., Menzel, L., Mitri, D.D., Drachsler, H.: Predicting item difficulty and item response time with scalar-mixed transformer encoder models and rational network regression heads. In: Workshop on Innovative Use of NLP for Building Educational Applications (2024), https://api.semanticscholar.org/CorpusID:270765450
14. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4302–4310. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
15. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D.M., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.: Learning to summarize from human feedback. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
16. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
17. van Otterlo, M., Wiering, M.: Reinforcement Learning and Markov Decision Processes, p. 3–42. Springer Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-27645-3_1, http://dx.doi.org/10.1007/978-3-642-27645-3_1
18. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023). https://doi.org/10.48550/ARXIV.2310.06825, https://arxiv.org/abs/2310.06825