

Hierarchical Attention Network Architecture for Automated Response Time Prediction

Wei Schneider
Cambium Assessment

Abstract

This paper presents a novel deep learning architecture for item response time prediction to create human readable feedback for content creators in educational assessment. The model employs a two-tier attention mechanism, mirroring the cognitive hierarchy of reading: the first tier identifies influential words nested within their sentence context, while the second tier evaluates the contribution of individual sentences to the overall item block. By leveraging this specific hierarchical structure, the model improves explainable AI (xAI) in educational measurement AI research. While architecturally simpler models can achieve comparable or superior raw predictive accuracy, the two-tier structure is designed specifically for xAI — to yield transparent, sentence- and world-level attribution that item developers can act on directly. This methodology bridges the gap between opaque, black-box predictive modeling and actionable, human interpretable feedback for instructional design and item development.

1. Introduction

Response time is a critical factor in educational assessment, directly impacting test fairness, validity, and cognitive fatigue (Bridgeman & Cline, 2000; Ackerman & Kanfer, 2009). Furthermore, predicting response time a priori is highly valuable, enabling content writers and test developers to better gauge item limits and pacing required for various test forms.

A growing body of literature has successfully leveraged large language models (LLMs) in prediction of item difficulty and response time (e.g., Baldwin, Yaneva, Mee, Clauser, & Ha, 2021; Veeramani, Thapa, Shankar, & Alwan, 2024), demonstrating the efficacy of natural

language processing techniques and pre-trained language models in psychometric modelling. However, many of these approaches often overlook the limitations inherent in these abstract models, which frequently operate as 'black boxes'.

One of the earlier deep learning models that was inherently a good candidate for xAI (Love et al., 2023) was the Hierarchical Attention Network (HAN; Yang et al., 2016) architecture. HAN introduces a two-tiered attention mechanism that models at both the word and sentence levels. By translating complex model weights into human-readable feedback, HAN explains its predictions by highlighting the specific words and sentences that drive its classification decisions. Such attention framework largely increases the transparency of the model but also could give practical feedback on how text features influence predicted feedback. This could be a very good use for item development as a feedback could pinpoint exactly where in an item text specific phrasing correlates with longer student response time.

The primary goal of this study is to leverage a hierarchical attention framework to construct an explainable deep learning model for educational assessment. Specifically, this research pursues a dual purpose: first, to assess the predictive accuracy of a two-tier attention architecture in forecasting student response times from raw item text; and second, to investigate the model's interpretability by examining whether its dual-level attention mechanisms yield actionable, human-readable feedback on textual complexity for item developers.

2. Methods

A sample of 656 middle school science items were selected for this study, comprising 342

standalone items and 314 cluster items. The comprehensive item texts--including both the stimulus and questions--was extracted from the assessment item bank through an Application Programming Interface (API). Due to API extraction limitations, embedded figures and tables were either incomplete or missing from the raw data dump. To maintain dataset integrity, systematic text preprocessing was applied to remove incomplete sentence fragments and partial tabular artifacts. To account for the omitted visual and structural components in the items that could affect response time, five proxy variables “number of figures”, “number of tables”, “number of equations”, “number of score points”, and “item type” were added to the dataset. The operational target variable for prediction was the 80th percentile of student response time (T_{80}), which was historically established as an administrative benchmark to distinguish regular length items from extra-long items. T_{80} serves as a metric for modeling item-level time intensity for this experimental study. The time distribution is shown in Figure 1. The distribution of response time is a right-skewed distribution where 5% of the entire distribution falls above 13.62 minutes. Notably, the distribution is bimodal, with distinct concentrations around 2-3 minutes and 8-10 minutes. This bimodality reflects the distinction between standalone and cluster items. Standalone items have a median response time of 2.70 minutes, while the cluster items have a median response time of 9.30 minutes.

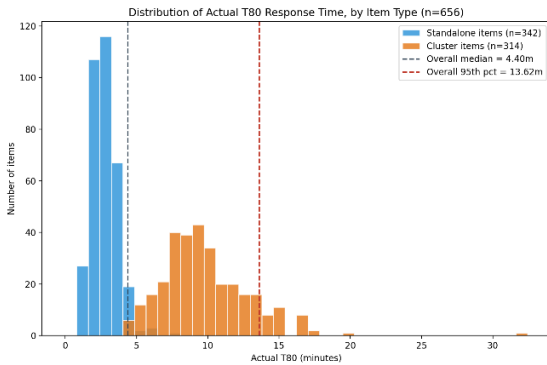


Figure 1. Response Time Distribution

A pre-trained language model “*electra-small-discriminator*” (Clark, Luong, Le & Manning,

2020) was used as the frozen feature extractor to produce informative token representations. With consideration of the limited sample size ($N = 656$), the low-rank adaptation (LoRA; Hu et al., 2022) was adopted to further fine-tune LLM efficiently. LoRA allows the encoder to adapt at every layer, while keeping the total number of trained parameters small. Two small trainable matrices, A and B , are introduced such that each layer’s output is computed as:

$$h = W * x + \left(\frac{\alpha}{r}\right) * B(A(x)) \quad (1)$$

where W is the original, pre-trained weight matrix inherited from the frozen language model; A is the first trainable LoRA matrix that projects the input down to a smaller low-rank subspace; B is the second LoRA matrix that projects the low-rank representation produced by A back into the original hidden dimension; r is the LoRA rank and α is a scaling factor.

The proposed model used a two-tier hierarchical attention mechanism (Figure 2) to estimate both the word-level and sentence-level feature importance. At the word level (Tier 1), each sentence was first encoded through *ELECTRA-small*, whose frozen weights were augmented with trainable LoRA matrices injected into the query and value projections of its 12 transformer layers. This yielded a contextualized embedding for each token with a dimension of 256. A learned projection maps to a 64-dimension attention space, from which a single scalar attention score is computed per token. These scores were then normalized via softmax within each sentence, consequently creating a set of word attention weights that sum to one per sentence. The sentence-level representation was then created as the weighted average of the original token embeddings (256 dimensions), using the word attention weights, which yield a distributed sentence representation across sentences.

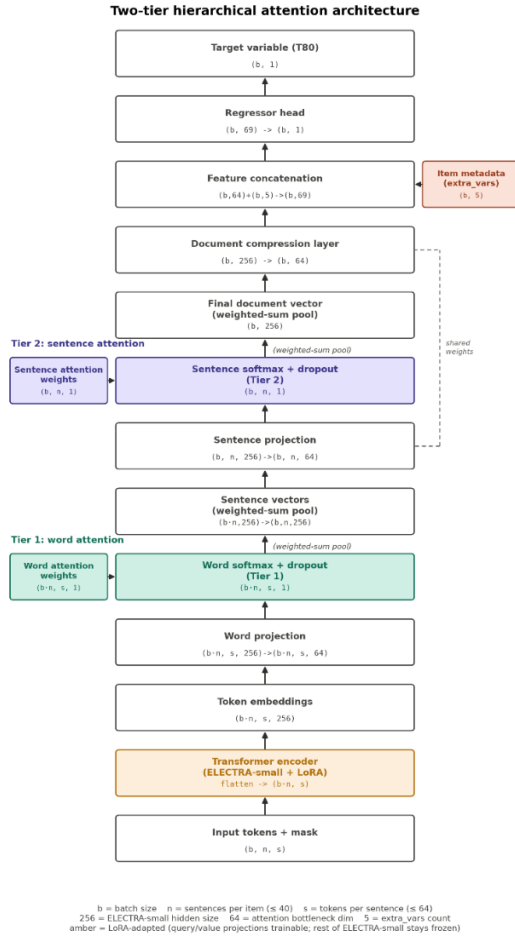


Figure 2. Two-Tier Attention Mechanism

At the sentence level (Tier 2), the sentence representations were treated analogously to the token embeddings of Tier 1, where each sentence representation was projected into a lower-dimensional attention space of 64, and passed through a scoring layer to produce one scalar attention score per sentence. Sentence representations were then pooled into document level as weighted average using sentence attention weights. Before the final regression head, the item attributes were concatenated with the compressed text representation to complement the textual information for the final prediction.

The two tiers were integrated sequentially rather than through an additional learned combination layer. This hierarchical structure

allowed the model to learn which words within a sentence, and which sentences within the item text, were most predictive of the target response-time outcome, while retaining information from all words and sentences at every stage of the computation. This serves as a key mechanism for highlighting important text for human review.

2.1 Baseline Models

The Flat Attention Baseline Model: The choice of a two-tier attention architecture as the primary model over a single flat attention mechanism is motivated by three key considerations. First, item text possesses an inherent multi-level structure; explicitly modeling both within-sentence word importance and across-sentence importance allows the network to represent this structure directly, rather than requiring it to be discovered implicitly by an undifferentiated attention layer. Second, the hierarchical design yields more interpretable output: the model produces both a sentence-level attention distribution and, within each sentence, a word-level distribution. This enables qualitative analyses that connect specific predictions to distinct sentence-and-word combinations within the item text. Third, because pre-trained language models have a fixed maximum input length, longer texts must be partitioned before encoding. The hierarchical model performs this partitioning at natural sentence boundaries, whereas a flat model cuts off text at arbitrary token limits, potentially splitting coherent semantic units across independently encoded chunks.

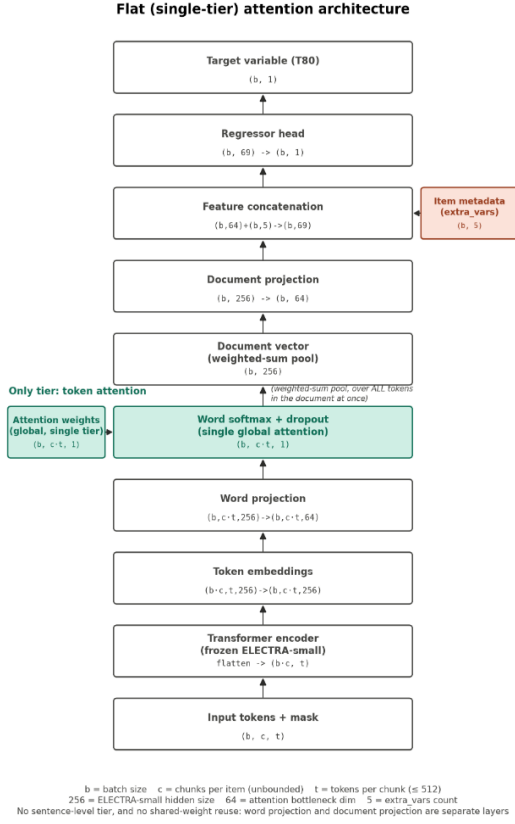


Figure 3. Flat Attention Baseline Model

To empirically evaluate these theoretical advantages, a flat single-level attention architecture was implemented as a primary baseline (Figure 3). This benchmark computes item-level predictions using a single attention layer applied over all tokens in the item text using the frozen *ELECTRA-small* embeddings. Because the encoder enforces a maximum sequence length of 512 tokens, the item text is divided into consecutive, fixed-length token chunks prior to modeling. Each chunk is encoded independently and then concatenating back into a single sequence spanning the full item text. A learned projection maps each token embedding in the reassembled sequence into a 64-dimensional space, where scalar attention scores are calculated and normalized through softmax across the entire document. The weighted sum of the token embeddings produces an aggregate text representation, which is projected into the same lower-dimensional space, concatenated with the

item attribute-features, and passed through the regression head to produce the final prediction.

XGBoost Baseline Model: To establish a benchmark evaluating the predictive capacity of the item attributes alone, a gradient-boosted tree model (XGBoost) was trained using only the five non-text item attributes as predictors: figure count, table count, equation count, score point count, and item type. Hyperparameters were fixed based on common defaults for small tabular datasets to achieve a reasonable baseline (Probst et al., 2019).

2.2 Model Training and Evaluation

Both the numeric item attribute variables and the target T_{80} values were normalized prior to training. Model performance was evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as primary metrics of predictive accuracy.

Both the two-tier and flat attention models were trained using the AdamW optimizer with a mean squared error (MSE) loss function. All models were implemented with PyTorch on the GPU and initialized with a fixed random seed to ensure exact reproducibility across experiments. To ensure generalizability of the calibrated, a nested cross-validation method was used to separate hyperparameter search from performance estimation. The full dataset ($N = 656$) was partitioned into 5 folds, with 1 fold being held out as the outer test set. For the inner loop, a further 80/20 split was used for training and validation purposes. Learning rates and epochs were learned through the training set with early stopping on the inner validation loss. The best learning rate and epoch count associated with it were retained to be tested on the outer fold, yielding that fold’s MAE and RMSE. This procedure was repeated for all five folds, and average statistics were reported as an estimate of the model’s expected performance on unseen items. A simple 5-fold cross-validation split was used to evaluate the XGBoost model.

Hyperparameter Tuning: To balance between model capacity and the risk of overfitting on such small sample, LoRA rank varied across $r = 0$ (i.e., a fully frozen ELECTRA-

small encoder, with no adapter matrices), $r = 4$, and $r = 8$.

A subset of hyperparameters within the two attention-based models were tuned to adapt to the specific characteristics of the item sample (Table 1), with tuning effort concentrated on learning rate and training duration (epochs) while others

were held constant. All models were trained for a maximum of 70 epochs, with an early stopping rule applied if validation loss failed to improve for 10 consecutive epochs.

Model	Learning Rate	Epoch	Parameters Fixed (Not Searched)
Flat (1-level attention)	2e-3, 1e-3, 5e-4, 2e-4, 1e-4	70 (via early stopping, patience=10)	maximum token chunk size = 512 per device train batch size = 16 per device eval batch size = 4 weight decay 0.01
Two-tier frozen ELECTRA ($r = 0$)	2e-3, 1e-3, 5e-4, 2e-4, 1e-4	70 (via early stopping, patience=10)	maximum sentence = 40 maximum words = 64 per device train batch size = 16 per device eval batch size = 4 weight decay 0.01 lora_r 0
Two-tier ELECTRA + LoRA ($r=4$)	2e-3, 1e-3, 5e-4, 2e-4, 1e-4	70 (via early stopping, patience=10)	maximum sentence = 40 maximum words = 64 per device train batch size = 16 per device eval batch size = 4 weight decay 0.01 lora_r 4 lora_alpha 8 lora_dropout 0.05
Two-tier ELECTRA + LoRA ($r=8$)	2e-3, 1e-3, 5e-4, 2e-4, 1e-4	70 (via early stopping, patience=10)	maximum sentence = 40 maximum words = 64 per_device_train_batch_size = 16 per_device_eval_batch_size = 4 weight_decay 0.01 lora_r 8 lora_alpha 16 lora_dropout 0.05

Table 1. Hyperparameters Fine-Tune by Model

3. Results

The fine-tuned learning rates and epochs per fold are presented in Table 2. Overall, the highest adopted learning rate was 0.0005 between the flat and two-tier frozen attention models, while 0.0010 became the more frequently selected value in LoRA rank 4 and 8 models. Overall,

XGBoost obtained the lowest errors across all models, with MAEs below 0.2 in most folds and RMSE below 0.3 in every fold. The flat attention model ranked second-best overall, with lower MAE and RMSE values than any of the two-tier attention model variants. The flat attention model also required the most epochs to train, while the

two-tier attention variants stayed below 21 epochs on average.

Overall, the accuracy scales inversely with model complexity. The more complexity and parameters added to the model, the worse the performance became. It strongly suggests that the larger models tend to overfit with small sample size. It is also shown in the fluctuation in the

accuracy indices across folds. For example, in the two-tier attention model with LoRA rank of 8, the MAEs range from 0.2893 to 0.4954 and RMSE going from 0.3589 to 0.5630, which indicates that extra model capacity is translating into instability rather than better fit.

Model	Outer Fold	Selected LR	Selected Epochs (Early Stop)	Fold MAE (log-space)	Fold RMSE (log-space)
Flat Attention Model	1	0.0005	44	0.2693	0.3355
	2	0.0005	35	0.2383	0.3086
	3	0.0005	29	0.2328	0.3005
	4	0.0002	69	0.1910	0.2534
	5	0.0005	38	0.2408	0.3117
Average			43.0	0.2345	0.3019
Two-tier Attention Model (frozen)	1	0.0005	34	0.2685	0.3276
	2	0.0005	39	0.3472	0.4270
	3	0.0005	37	0.2394	0.3085
	4	0.0005	31	0.2722	0.3361
	5	0.0005	30	0.2604	0.3248
Average			34.2	0.2775	0.3448
Two-tier Attention Model (LoRA Rank = 4)	1	0.0010	15	0.2677	0.3416
	2	0.0010	14	0.2758	0.3449
	3	0.0005	27	0.2797	0.3427
	4	0.0005	18	0.3594	0.4230
	5	0.0010	18	0.3259	0.4078
Average			18.4	0.3017	0.3720
Two-tier Attention Model (LoRA Rank = 8)	1	0.0005	38	0.2893	0.3715
	2	0.0010	15	0.4954	0.5630
	3	0.0010	10	0.3242	0.3936
	4	0.0005	20	0.3002	0.3589
	5	0.0010	21	0.3201	0.4044
Average			20.8	0.3458	0.4183
XGBoost (attribute-only)	1	N/A	N/A	0.2159	0.2793
	2	N/A	N/A	0.1774	0.2312
	3	N/A	N/A	0.1759	0.2183
	4	N/A	N/A	0.1931	0.2497
	5	N/A	N/A	0.1741	0.2272

Average				0.1873	0.2411
---------	--	--	--	--------	--------

Table 2. Model Performance by Fold

Figure 4 shows the scatterplots of predicted and actual 80th percentile time across 5 models. The XGBoost’s prediction show a distinct gap between 4 to 7 minutes, which mirrors the bimodality of the actual response time shown in Figure 1. The attention-based models show a more continuous, blended spread of predictions across the same region, likely because these models only have indirect access to the item type through text rather than as an explicit feature.

Among the attention-based models, the flat attention-based model was the most symmetric in terms of over- and under-estimates. Prediction error increases for all models as actual response time grows, and this effect is more pronounced for the attention-based models than for XGBoost. Notably, the LoRA rank-8 model shows a more severe tendency to overestimate response time than other models.

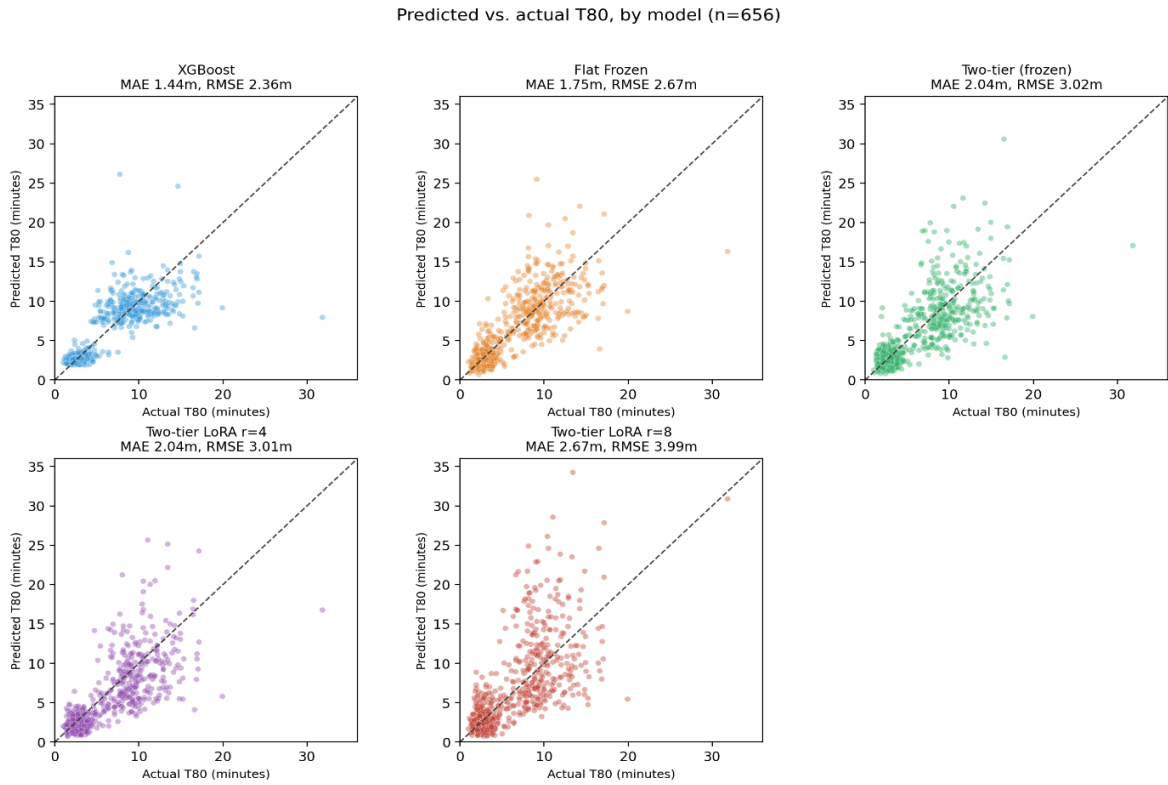


Figure 4. Scatterplots of Predicted and Actual 80th Percentile Response Time

Table 3 reports the weight distribution statistics and word type percentages. The weights are pooled across all words produced by each model. Because the flat model spreads attention across entire item while the two-tier models spread attention within one sentence at a time, the resulting word-level weights cannot be directly

compared. We examined each item’s single highest-weighted word, identified by the flat and two-iter attention-based models, to assess whether the item’s structural characteristics were being captured through attention. If the identified word contains no letters or digits at all, it is counted as punctuation. If it is a noun, verb, or

adjective, it is counted as a content word. Anything else is considered a function word. The flat attention-based model had the highest percentage of all on punctuation (33.2%), while the two-tier model with LoRA rank 4 had the lowest (7.3%). For all models, content words were the most common and punctuation words were the least common, indicating that the

models functioned as expected to put emphasis on meaningful words. Once LoRA was applied, the attention shifted markedly toward content words and function words, and punctuation dropped. LoRA $r = 4$ achieved the lowest punctuation share of all models, while LoRA $r = 8$ showed the largest increase in content-word attention.

Model	Mean Word-level Weight	SD	% Punctuation	% Function Word	% Content Word
Flat (frozen)	0.00265	0.00292	33.2	16.0	50.8
Two-tier (frozen)	0.04761	0.03804	14.5	23.8	61.7
Two-tier LoRA $r=4$	0.05000	0.04470	7.3	27.4	65.2
Two-tier LoRA $r=8$	0.04762	0.04301	9.0	12.7	78.4

Table 3. Word Level Weights Statistics and Word Type Importance Percentages

Taken together, these findings support the paper’s central framing: the two-tier attention architecture’s value lies in interpretability. Adapting the encoder with LoRA further transformed what attention identifies as important, without sacrificing predictive accuracy at $r = 4$. This suggests that even minimal, parameter-efficient adaptation of the encoder is sufficient to redirect the model’s attention towards linguistically meaningful content.

The primary strength of the proposed model lies in its capacity for exporting word- and sentence-level weights that can map back onto the raw item text for human interpretability. These weights serve as diagnostic output, which can be rendered into a highly accessible format. By visually highlighting the core friction words in a sentence, the model provides granular, text-based insights. This interpretative capability creates a continuous feedback loop that enables content writers to engage in—making targeted adjustments and immediate evaluation of the predicted impact on response time.

4. Conclusions and Future Work

In this paper, the findings of this study demonstrate the viability of using the two-tier

attention architecture to model student response times and provide interpretable diagnostic feedback for item development. To build upon these results, future research should expand the item sample size across a broader range of academic domains, such as English Language Arts assessments, where reading comprehension drives the majority of response latency.

Two limitations motivate future work with a larger item sample. First, LoRA rank 8 did not improve on rank 4 — it showed greater fold-to-fold instability and lower accuracy, indicating that this dataset supports only a small amount of additional encoder capacity before overfitting risk outweighs any benefit. Second, sentence-level attention remained close to uniform across sentences within an item, indicating limited ability to discriminate important sentences from unimportant ones; this will also need further investigation with a larger sample. Additionally, future iterations of this work will focus on integrating true multimodal architectures capable of directly processing embedded figures, tables, and mathematical equations, rather than relying on scalar counts of these elements as proxy features.

5. Acknowledgements

I thank my colleagues Claire Chang, Suhwa Han, Zebing Wu, and Mina Lee for their earlier work on the topic and helpful discussions around this work.

References

- Ackerman, P. L., & Kanfer, R. (2009). Test length and cognitive fatigue: an empirical examination of effects on performance and test-taker reactions. *Journal of Experimental Psychology: Applied*, 15(2), 163. <https://doi.org/10.1037/a0015719>
- Phillip L. Ackerman and Ruth Kanfer. 2009. Test length and cognitive fatigue: An empirical examination of effects on performance and test-taker reactions. *Journal of Experimental Psychology: Applied*, 15(2):163–181. <https://doi.org/10.1037/a0015719>
- Baldwin, P., Yaneva, V., Mee, J., Clauser, B. E., & Ha, L. A. (2021). Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1), 4–30.
- Peter Baldwin, Victoria Yaneva, John Mee, Brian E. Clauser, and Le An Ha. 2021. Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, 58(1):4–30. <https://doi.org/10.1111/jedm.12264>
- Brent Bridgeman and Frederick Cline. 2000. Variations in mean response times for questions on the computer-adaptive GRE® general test: Implications for fair assessment. *ETS Research Report Series*, 2000(1):i–29. <https://doi.org/10.1002/j.2333-8504.2000.tb01830.x>
- Bridgeman, B., & Cline, F. (2000). Variations in mean response times for questions on the computer-adaptive GRE® general test: implications for fair assessment. *ETS Research Report Series*, 2000(1), i–29.
- Clark, K., Luong, M., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2003.10555>
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hariram Veeramani, Surendrabikram Thapa, Natarajan Balaji Shankar, and Abeer Alwan. 2024. Large Language Model-based Pipeline for Item Difficulty and Response Time Estimation for Educational Assessments. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 561–566, Mexico City, Mexico. Association for Computational Linguistics. <https://aclanthology.org/2024.bea-1.49/>
- Kevin Clark, Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.2003.10555>
- Peter E. D. Love, Weili Fang, Jane Matthews, Stuart Porter, Hanbin Luo, and Lieyun Ding. 2023. Explainable artificial intelligence (XAI): Precepts, models, and opportunities for research in construction. *Advanced Engineering Informatics*, 57:102024.
- Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. 2019. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20(53):1–32.
- Veeramani, H., Thapa, S., Shankar, N. B., & Alwan, A. (2024). Large language model-based pipeline for item difficulty and response time estimation for educational assessments. *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)* (pp. 561–566). <https://aclanthology.org/2024.bea-1.49/>
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1480–1489). San Diego, California. Association for Computational Linguistics. <https://doi.org/10.18653/v1/N16-1174>
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics. <https://doi.org/10.18653/v1/N16-1174>