

Production-Ready Automated Item Generation in Educational Assessment: Integration with Operational Workflows

Hotaka Maeda ^{1,2*}

Kargi Chauhan ²

Tharunya Chandrashekar¹

¹Smarter Balanced, Santa Cruz, California, United States of America

²University of California-Santa Cruz, Santa Cruz, California, United States of America

Abstract

We present a production-ready automated item generation pipeline integrated into a large-scale assessment program's workflows. A one-shot approach prompts LLMs from an operational source item and existing guidelines, then populates metadata, screens quality, and uploads items for human review. Generated items passed expert review, and difficulty prompting reliably shifted difficulty.

1 Background

Large-scale educational assessments require hundreds or thousands of new items annually, yet traditional item development remains slow and expensive. Recent work has shown that automated item generation (AIG) using large language models (LLMs) can produce structurally plausible items across a range of subjects and formats (Circi et al., 2023; Falcão et al., 2022; Tan et al., 2025; Omojekunola et al., In press). However, most AIG research treats generation as a standalone task, producing items in isolation from the testing program that would ultimately use them. A production-ready AIG pipeline instead requires operational integration. This is challenging when the generation process must be embedded into a preexisting examination program that has run for years and already has its own item requirements, data, platforms, and human workflows. Established programs have guidelines defining what items must look like, an item bank database holding operational items, and a content team that writes and revises items through interfaces they use daily.

This challenge is not unique to assessment. Organizations broadly have struggled to translate AI adoption into realized value, a pattern often described as the AI productivity paradox (McElheran

et al., 2025; Brynjolfsson et al., 2021, 2017). Evidence consistently points to organizational and process factors rather than AI model capability as the binding constraint. The challenge is acute in established organizations whose existing routines, legacy systems, and layered processes are hard to adapt (Brynjolfsson et al., 2021). Educational testing programs are a clear example of this setting.

Framing production-readiness as integration has concrete consequences for how the pipeline is built. First, because the program already maintains documented item requirements, these can be fed directly into LLMs at generation time rather than reinvented. Second, because the program already has a bank of operational items, a single existing item can serve as a one-shot source item, which lets generated items inherit the source item's metadata rather than requiring it to be reconstructed. Third, because items are stored and reviewed in an existing database, generated items can be imported back into that same system, keeping data organized and letting content experts review new items in a familiar platform. Fourth, because items are stored as code rather than rendered form, this round trip through the database is what renders them into the visual format human reviewers expect (see Figure 1). Together, these design choices let an established program adopt AIG with minimal disruption and added effort while still holding items to its existing standards.

2 Aims

We present an AIG pipeline for established large-scale standardized English Language Arts (ELA) and mathematics assessments designed for production use. Rather than treating generation as a standalone exercise, the system is designed to fit the infrastructure, policies, and people of the Smarter Balanced testing program. After introducing the pipeline, we evaluate it in two studies. Study 1 is a

*Correspondence concerning this article should be addressed to Hotaka Maeda, Smarter Balanced, University of California-Santa Cruz, 1156 High St, Santa Cruz, CA 95064. E-mail: hotaka.maeda@smarterbalanced.org

(a) *Generated JSON*

```
"options": [
  {
    "isCorrectAnswerKey": false,
    "text": "<p>81</p>"
  },
  {
    "isCorrectAnswerKey": true,
    "text": "<p>108</p>"
  },
  {
    "isCorrectAnswerKey": false,
    "text": "<p>112</p>"
  },
  {
    "isCorrectAnswerKey": false,
    "text": "<p>120</p>"
  }
],
"prompt": "<p>The class collects 12 cans on Monday and 15 cans on Tuesday
. They collect the same total each week for 4 weeks.</p><p><br/>
How many cans do they collect in 4 weeks?"
```

(b) *Rendered item*

The class collects 12 cans on Monday and 15 cans on Tuesday.
They collect the same total each week for 4 weeks.

How many cans do they collect in 4 weeks?

- (A) 81
- (B) 108
- (C) 112
- (D) 120

Figure 1: Example generated multiple choice item. The large language model generates items in HTML format embedded within a JSON format. Once uploaded to the item bank database, the rendered item can be viewed for human review on the front-end interface of the database, the same platform the content team already uses to write and revise items. Only a portion of the generated JSON is shown.

small initial check of whether the generated items are acceptable to human content experts. Study 2 uses a larger generated item pool to examine how the generation inputs affect item quality. We vary the content specifications, fairness guidelines, and target difficulty, and evaluate the resulting items with automated screening and prediction methods.

3 Proposed AIG Pipeline

The pipeline is organized into seven modular sequential stages.

1. **Data retrieval:** We fetch a manually selected operational item from a database using queries. This source item plays a role analogous to the item model in traditional automated item generation (Gierl and Lai, 2012), defining the structure, style, metadata, and covered content standard for the newly generated items. Unlike a conventional item model,

which is a manually engineered template, the source item is an existing operational item used directly as a one-shot example. We also retrieve related item development documents, including the content specifications and sensitivity and fairness guidelines that govern the existing testing program.

2. **One-shot prompt construction:** All retrieved information is converted into a prompt. HTML format embedded within JSON is used, which is both well suited for LLM processing and consistent with the format used to store items in the item database. The LLM is instructed to mirror the organization and conventions of existing items by following the structured JSON format of the one-shot source item, and instructed to return its response in the same structured format (see Appendix A). Using the database’s own storage format at both input and output is what allows generated items to move back into the operational system later without reformatting.
3. **Item generation:** Send the prompt to OpenAI GPT-5.6 Terra through an API (OpenAI, 2026). The LLM generates text content of multiple items alongside scoring rubrics in a structured JSON format.
4. **Metadata Assignment:** The one-shot approach allows the LLM to mimic many features of the source item, which is an advantage because the metadata of the source item can be deterministically copied to the generated items. While items can have hundreds of back-end metadata fields, examples include the subject, grade level, blueprint categories, standard alignment, or item type. This significantly reduces the burden of post-generation item data management. Note that not all metadata and auxiliary data can be copied because they would be different for the new items (e.g., item parameters or text-to-speech audio data).
5. **Natural language processing (NLP) quality control (QC):** We compare generated items against the known distribution of operational item characteristics using ten NLP features computed from item text using the textstat Python package (Bansal and Aggarwal, 2025). These included word count, letter count, sentence count, words per sentence, average syllable

bles per word, average letters per word, Flesch Reading Ease, Automated Readability Index (ARI), Dale-Chall readability score, and estimated reading time. Specifically, we confirm that these variables are within the minimum and maximum range observed in operational items within the same subject, grade, item type, and claim. Items that do not meet the required ranges are rejected outright.

6. **AI review:** Similar prompt from Step 2 where the source items are replaced with the generated items are sent to GPT-5.6 Terra again to re-check the fidelity to all requirements (see [Appendix B](#)). The LLM classified each item as "accept", "flag", or "reject", and generated a short rationale. The returned responses are provided to the content experts to help assist during the human review phase.
7. **Item bank database upload:** Upload accepted items to the secure item database environment along with the metadata (i.e., the same database that stores the source item) through its API. Successful upload serves an additional quality assurance check, as the data format must match what is expected by the database. Critically, items are stored as code rather than in rendered form, so importing them back into the database is what renders them into the visual format reviewers expect. Uploaded items are available for immediate render and human review on the front-end interface of the database, the same platform the content team already uses to write and revise items.

4 Study 1: Initial Acceptability

An initial small study was conducted to ensure that the AIG system was functioning acceptably well prior to a more in-depth evaluation. Four different types of source items were used for the study: (1) ELA multiple-select (MS) and (2) hot-text-question-select (HTQS; selection of words within paragraphs), and (3) math multiple-choice (MC) and (4) equations. The four source items were used to generate ten items each ($4 \times 10 = 40$ items) and were sent to the ELA and math content experts for human review.

5 Study 1: Results

A total of 8/10 ELA multiple-select items, 10/10 hot-text-question-select items, 6/10 Math multiple-choice, and 10/10 equation items passed NLP QC. ELA and math content experts reviewed five items for each of the item types ($4 \times 5 = 20$ items). They all passed human review by three reviewers per subject, and were therefore all usable for operations with little to no modification. However, a key challenge was that flaws in source items were sometimes inherited by the generated items. This makes careful source item selection essential, as well as revision of any flawed source items before generation begins. LLM usage costs were less than 2 cents per generated item.

6 Study 2: Ablations and Item Difficulty

Study 1 demonstrated that the AIG pipeline can produce operationally viable items integrated into existing workflows. Study 2 examines how the two primary guideline inputs, content specifications and fairness and sensitivity guidelines, along with difficulty prompting, affect the quality and linguistic characteristics of generated items. We generated a larger item pool under systematically varied conditions, characterized the items using NLP, and evaluated their operational viability using prediction models.

6.1 Item Generation Design

We used a fully crossed experimental design with three factors. The first factor was the inclusion of content specifications, either present or absent. The second factor was the inclusion of the fairness and sensitivity guidelines, either present or absent. The third factor was the target difficulty, where the prompt specified to generate items easier, similar, or harder difficulty compared to the source item. Crossing these factors produced $2 \times 2 \times 3 = 12$ conditions. When one or both guideline inputs were absent, the corresponding portion of the prompt was removed entirely.

Ten operational source items served as one-shot anchors, five for ELA (all grade 3, 3 HTQS, 1 MC, 1 MS item) and five for mathematics (all grade 4, 3 MC, 2 equations). Each source item was used to generate items under all 12 conditions, and 10 items were generated per source item within each condition. This yielded $12 \times 10 \times 10 = 1,200$ generated items in total, balanced across the two subjects. All items were generated using the pro-

posed pipeline, with the difficulty instruction and the guideline ablations applied at the prompt construction stage.

6.2 Three Automated Screening Criteria

We characterized the linguistic patterns of generated items using ten NLP features computed from item text in the NLP QC step. As described in the proposed pipeline, items were labeled as "reject" when any of these variables were outside of the minimum and maximum range observed in operational items within the same subject, grade, item type, and claim. In addition, AI reviewer results are analyzed by treating "accept" and "flag" items as "accept", and rejecting the "reject" items.

To assess the operational viability of the generated items, we applied the automated item evaluation (AIE) fusion model (Maeda and Lu, 2026a) to predict whether an item would be accepted or rejected for operational use. It was trained on 52,759 ELA and mathematics items from the same testing program targeted in the current paper. The fusion model combines two fine-tuned DeBERTaV3-large encoders (He et al., 2021). The first encoder processes the raw item text. The second encoder processes a two-sentence item quality critique generated by a Qwen3-0.6B language model (Qwen Team, 2025). The 1,024-dimensional final-layer representations from both encoders are concatenated and passed through a fusion head to produce a rejection probability. On held-out test data, predictions achieved stronger performance for math ($AUC = .86$) than for ELA ($AUC = .72$).

The AIE model was used to produce a rejection probability for all 1,200 generated items. Probabilities at or above .25 (rather than .50) were treated as rejected. This low threshold has a high sensitivity (.88 for ELA and .91 for math) with reduced specificity (.31 for ELA and .56 for math), which is recommended in an AIG context where generating additional items has a low cost (Maeda and Lu, 2026a).

6.3 Predicting Item Discrimination and Difficulty

To assess whether difficulty prompting shifted item difficulty properly, and whether the automated screening methods reject items with low discrimination, we predicted item discrimination (a) and difficulty (b) using a prediction model (Maeda and Lu, 2026b). We use the text-only variant of the model (i.e., no images and no item-level metadata

input), which was a fine-tuned DeBERTaV3-large (He et al., 2021) that converts each item text component into a 128-dimensional representation, aggregates these into an item-level representation using attention pooling, and predicts item response theory item parameters through a regression head. Trained on the same standardized test data targeted in the current paper, predicted discrimination and difficulty achieved R^2 values of .45 and .38, respectively.

6.4 Statistical Analysis

We tested whether the experimental conditions affected the generated items using mixed-effects models fit in R with the lme4 package (Bates et al., 2015). Each generated item was nested within the source item it was derived from, so all models included a random intercept for source item to account for the non-independence of items sharing an anchor. The three experimental factors were entered as fixed effects. Continuous outcomes, including the NLP metrics, predicted item difficulty, and predicted discrimination, were analyzed with linear mixed models. The three binary screening criteria (NLP, AI reviewer, and AIE model) were analyzed with a generalized linear mixed model using a logit link and a binomial response distribution. All models were estimated by maximum likelihood so that models differing in their fixed-effects structure could be compared directly.

To determine whether the experimental factors affected the outcomes, using likelihood ratio tests, we compared the (1) full model with all main effects and 3-way interactions, (2) reduced model with only main effects, and (3) a null model with only the intercept. Fixed-effect significance ($\alpha = .05$) was assessed using Satterthwaite-approximated degrees of freedom for the linear models via the lmerTest package (Kuznetsova et al., 2017). NLP metrics were compared only across the difficulty condition. Spearman correlations between the screening criteria and predicted item discrimination are also reported. When calculating correlations with AI reviewer decisions, all three categories were used (accept=0, flag=1, reject=2).

7 Study 2: Results

7.1 Screening Evaluation

We applied three independent screening criteria to the generated items. With a mean predicted rejection

tion probability of .35 ($SD = .20$), 75% of ELA and 50% of math items were rejected by the AIE model. These figures are only 6 percentage points above the model's false positive rate, which provides partial evidence that these items are generally good quality. The AI reviewer rejected 46% of items, and NLP QC rejected 16%. Only 13% of ELA and 22% of math items passed all three criteria. NLP and AI review were modestly related ($\rho = .19$), while the AIE model was essentially uncorrelated with both NLP ($\rho = .06$) and AI review ($\rho = -.05$). These low correlations show that the three screening tools are not redundant and each may catch distinct item issues. The AI reviewer ($\rho = -.44$), NLP ($\rho = -.20$), and AIE ($\rho = -.11$) screening criteria all tended to reject items that had low predicted item discrimination.

Interactions effects were not present for the NLP method, $\chi^2(7) = 2.7, p = .91$, nor AI reviewer, $\chi^2(7) = 1.91, p = .96$, while all three main effects were significant for both ($p < .05$) except for the bias guideline effects for the NLP method ($p = .75$). Predicted AIE rejection rates did not differ across any experimental condition, $\chi^2(11) = 4.5, p = .95$.

Rejection rates differed markedly by subject. The AI reviewer rejected math items (61%) far more often than ELA items (32%). The NLP-based QC showed the reverse, rejecting more ELA items (21%) than math items (11%). For ELA, supplying the content specifications decreased AI reviewer rejection rates from 38% to 25%, and NLP rejection rate from 24% to 17%. AI reviewer rejected more ELA items when items were prompted to be easier (40%) than similar (31%) or harder (24%). The bias guidelines slightly raised the ELA rejection rate (37% versus 26%) rather than lowering it. For math, AI reviewer and NLP rejection rates were similar across all conditions. The only exception was that the NLP method tended to flag harder (14%) and similar (14%) math items than easier ones (6%).

7.2 NLP Metrics

All ten NLP metrics were statistically significantly affected by the difficulty conditions ($p < .05$). The generated items' linguistic features shifted with prompted difficulty, providing convergent evidence that the difficulty manipulation operated as intended (see Figure 2). For mathematics, mean word count rose from 39 to 44 to 47 words across the easier, similar, and harder conditions, while

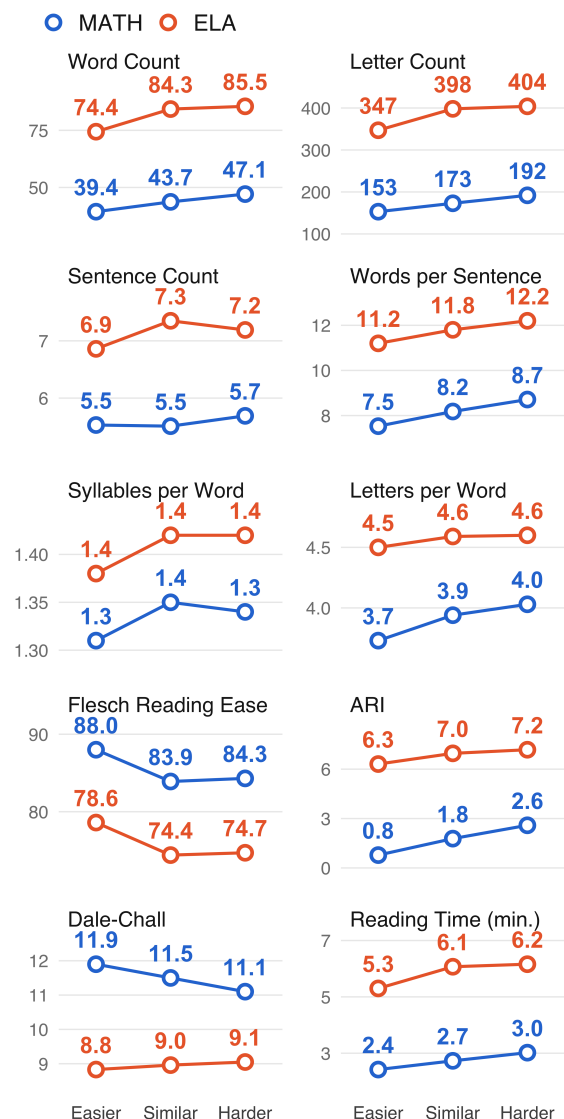


Figure 2: NLP metrics by difficulty prompting condition. NLP = natural language processing, ELA = English language arts, ARI = Automated Readability Index. Prompt specified to generate items easier, similar, or harder difficulty compared to the source item.

for ELA it rose from 75 to 84 to 86 words. The ARI showed a corresponding rise from 0.8 to 2.6 for mathematics and from 6.4 to 7.2 for ELA. Letter count, words per sentence, and Flesch Reading Ease all moved in the expected direction for both subjects as well, indicating that the difficulty effect reflects a genuine change in item complexity rather than an artifact of any one measure. For several metrics the largest step occurred between the easy and medium conditions, with little further change from medium to hard. The only unexpected result was the Dale-Chall score for mathematics, which indicated simpler vocabulary in harder items. However, Dale-Chall is designed to quantify reading

comprehension difficulty, and may not be effective for gauging math item difficulty.

7.3 Predicted Item Discrimination and Difficulty

Interactions effects of the conditions were not present for item discrimination, $\chi^2(7) = 9.0$, $p = .25$, nor difficulty, $\chi^2(7) = 8.04$, $p = .33$, while all three main effects were significant for both ($p < .01$). Items had 0.06 higher predicted discrimination when prompted for easier difficulty compared to harder. Difficulty prompting shifted predicted item difficulty in the intended direction (see Figure 3). Items prompted to be easier, similar difficulty, or harder than the source items averaged -0.85, -0.55, and 0.20 for math, and -0.99, -0.72, and -0.69 for ELA, respectively (see Table 1). Difficulty prompting alone accounted for 10.0% of the variance in predicted difficulty.

Generated items prompted to be similar difficulty, $M(SD) = -0.75(0.50)$, tended to be more difficult than their source items $-1.07(0.42)$. Prompting for easier items, $-1.01(0.40)$, produced the closest match to source item difficulty. Items became modestly less discriminating and more difficult when content specifications and bias guidelines were supplied ($p < .01$). This is consistent with the expectation that adding constraints to the generation prompt can trade off against other item qualities.

8 Discussion

This study presents a successful end-to-end AIG pipeline designed for production use that adapts to ELA and mathematics configurations within the Smarter Balanced assessment program. The key design decision is that our AIG system takes advantage of the item development structures that were already in place. The existing item bank platform, source items, content specifications, and sensitivity and fairness guidelines provide a backbone to generating operationally viable items. We also used prediction models (Maeda and Lu, 2026a,b) trained on historical assessment program data to evaluate the generated items. Using these models, NLP, and AI review, newly generated items are automatically vetted. Items with high chance of being good quality are uploaded to the item bank, where they are rendered and immediately available for human review. The result is a smooth transition from machine generation to human review. Our system is

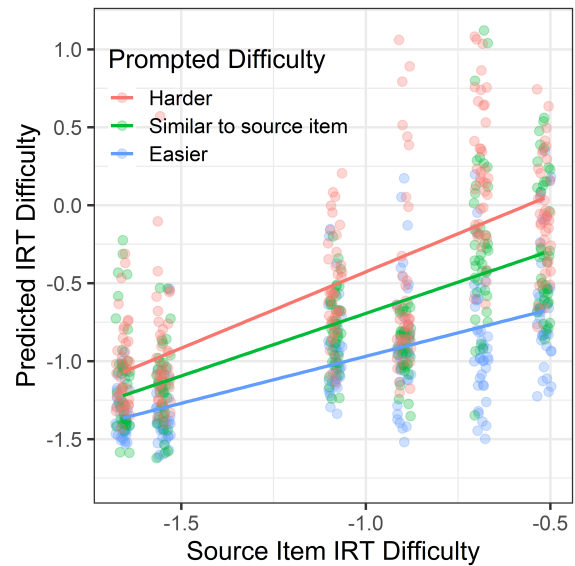


Figure 3: Predicted difficulty of generated items relative to the source item difficulty and prompt. IRT = item response theory. Predicted IRT b-parameters based on the 2-parameter logistic model are from a text-based prediction model (Maeda and Lu, 2026b). Shown are 721 items generated from 6 of the 10 source items that had item parameters.

also highly adaptable and modular. For example, we generated four different item types, while many AIG systems are designed to generate only multiple choice items (Gierl et al., 2012; Song et al., 2025). Our system is item-type agnostic, relying only on the source item and content specification. This design lowers the friction of transitioning an assessment program to adopt an AIG system.

The three automated screening criteria all tended to reject items with poor predicted item discrimination, but the criteria were not redundant. Each appears to capture a distinct class of item issues. The NLP-based QC method catches objective deviations from the operational norm, such as excessive length. The AI reviewer was more strongly associated with predicted item discrimination, suggesting it may identify items that are not designed cohesively, such as response options that do not function well together. The AIE model was trained to reproduce the testing organization’s historical accept and reject decisions, so it arguably rejects items for the broadest range of reasons among the three. Because the criteria are complementary rather than overlapping, all three could be incorporated into the QC steps of the AIG pipeline.

Difficulty prompting effectively controlled the predicted difficulty of the generated items. Change

Factor	Category	ELA		Math	
		Discrimination	Difficulty	Discrimination	Difficulty
Content specs	Absent	0.67 (0.10)	−0.87 (0.63)	0.72 (0.17)	−0.56 (0.54)
	Present	0.63 (0.12)	−0.73 (0.65)	0.72 (0.18)	−0.48 (0.54)
Sensitivity and Fairness Guide	Absent	0.66 (0.10)	−0.84 (0.60)	0.73 (0.18)	−0.55 (0.56)
	Present	0.64 (0.13)	−0.76 (0.68)	0.72 (0.17)	−0.50 (0.51)
Difficulty	Easier	0.70 (0.11)	−0.99 (0.61)	0.74 (0.15)	−0.85 (0.38)
	Similar	0.64 (0.12)	−0.72 (0.65)	0.73 (0.17)	−0.53 (0.49)
	Harder	0.62 (0.11)	−0.69 (0.63)	0.70 (0.20)	−0.20 (0.53)

Table 1: Predicted discrimination and difficulty by condition and subject. ELA = English language arts. Values are M (SD) of 2-parameter logistic model parameters, predicted using a model introduced by Maeda and Lu (2026b). Higher b indicates a more difficult item. Main effects of each factor was statistically significant ($p < .01$).

was also evident in all linguistic features of the items, which provide convergent evidence that the difficulty manipulation worked as intended. One consistent departure was that generated items tended to be slightly more difficult than the source item even when prompted for similar difficulty.

Supplying content specifications lowered ELA rejection rates, indicating that faithfulness of item structure can be reviewed reasonably well with a machine-based method. However, whether the fairness guidelines improved the items was less clear. Their effect was difficult to assess with any of the three automated screening criteria, and including them did not meaningfully change screening rates. This leaves two possibilities. Either the guidelines did not substantially change the generated items, or the screening methods were unable to detect the changes they produced. The latter is plausible given that the AIE model detects bias, sensitivity, and fairness concerns poorly (Maeda and Lu, 2026a). Our field consistently warns that we cannot place blind trust in AI-generated content in educational measurement (Bulut et al., 2024; Yin et al., 2025). Human review should therefore follow every AIG process, with reviewers concentrating their attention on the fairness, bias, and sensitivity issues that machines handle least reliably. All items from Study 1 passed human review, providing some evidence that our system is typically able to avoid these concerns.

A key limitation of our pipeline is the inability to generate complex items such as those involving images, graphs, tables, or testlets, and the reliance on high-quality preexisting source items, which prevents the generation of entirely new item types. Given the modular design of the pipeline, several of these shortcomings could be addressed through fur-

ther development. Another limitation was that only 6 of the 10 source items (4 mathematics and 2 ELA) had operational item parameter data available for analysis. Finally, the difficulty and discrimination comparisons depend on predicted item parameters, which should be taken with caution.

Acknowledgments

We gratefully acknowledge the contributions of Alec Gross, Alex Dean, Ana Stanciu, Aren Simmons, Brad Vasil, Catherine Francis, Danielle Ricks, Gilbert Salazar, Katie Westgate, Natasha Johnson, Shumin Jing, and Valerie Pannell.

References

- Shivam Bansal and Chaitanya Aggarwal. 2025. [textstat](#).
- Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. [Fitting linear mixed-effects models using lme4](#). *Journal of Statistical Software*, 67(1):1–48.
- Erik Brynjolfsson, Daniel Rock, and Chad Syverson. 2017. [Artificial intelligence and the modern productivity paradox: A clash of expectations and statistics](#). NBER Working Paper 24001, National Bureau of Economic Research, Cambridge, MA.
- Erik Brynjolfsson, Daniel Rock, and Chad Syverson. 2021. [The productivity J-curve: How intangibles complement general purpose technologies](#). *American Economic Journal: Macroeconomics*, 13(1):333–372.
- Okan Bulut, Maggie Beiting-Parrish, Jodi M Casabianca, Sharon C Slater, Hong Jiao, Dan Song, Christopher M Ormerod, Deborah Gbemisola Fabiyi, Rodica Ivan, Cole Walsh, et al. 2024. [The rise of artificial intelligence in educational measurement: Opportunities and ethical challenges](#). *Chinese/English Journal of Educational Measurement and Evaluation*, 5(3).

- Ruhan Circi, Juanita Hicks, and Emmanuel Sikali. 2023. Automatic item generation: foundations and machine learning-based approaches for assessments. *Frontiers in Education*, 8.
- Filipe Falcão, Patrício Costa, and José Miguel Pêgo. 2022. Feasibility assurance: a review of automatic item generation in medical assessment. *Advances in Health Sciences Education*, 27(2):405–425.
- Mark J. Gierl and Hollis Lai. 2012. [The role of item models in automatic item generation](#). *International Journal of Testing*, 12(3):273–298.
- Mark J Gierl, Hollis Lai, and Simon R Turner. 2012. Using automatic item generation to create multiple-choice test items. *Medical education*, 46(8):757–765.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *arXiv preprint arXiv:2111.09543*.
- Alexandra Kuznetsova, Per Bruun Brockhoff, and Rune Haubo Bojesen Christensen. 2017. [lmerTest package: Tests in linear mixed effects models](#). *Journal of Statistical Software*, 82(13):1–26.
- Hotaka Maeda and Yikai Lu. 2026a. Automated item evaluation: Predicting item acceptance and rejection using LLM-generated critiques. *arXiv preprint arXiv:2608.06609*.
- Hotaka Maeda and Yikai Lu. 2026b. [Multimodal test item parameter prediction from text, images, and metadata: Fusing together AI vision and language models](#). *Educational and Psychological Measurement*.
- Kristina McElheran, Mu-Jeung Yang, Zachary Kroff, and Erik Brynjolfsson. 2025. *The Rise of Industrial AI in America: Microfoundations of the Productivity J-curve (s)*. US Census Bureau, Center for Economic Studies Suitland, MD, USA.
- Moses. O Omopekunola, Anna. M. Gorbacheva, and Monday. A. Onoja. In press. Evaluating LLM-generated assessment items: A JBI-guided critical appraisal and checklist. *International Journal of Evaluation and Research in Education*.
- OpenAI. 2026. GPT-5.6 terra model documentation. <https://developers.openai.com/api/docs/models/gpt-5.6-terra>. Accessed 2026-08-09.
- Qwen Team. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- Yishen Song, Junlei Du, and Qinhua Zheng. 2025. [Automatic item generation for educational assessments: a systematic literature review](#). *Interactive Learning Environments*, 33(9):5386–5405.
- Bin Tan, Nour Armoush, Elisabetta Mazzullo, Okan Bulut, and Mark J. Gierl. 2025. A review of automatic item generation techniques leveraging large language models. *International Journal of Assessment Tools in Education*, 12(2):317–340.
- Zhipeng Yin, Sribala Vidyadhari Chinta, Zichong Wang, Matthew Gonzalez, and Wenbin Zhang. 2025. [Fairaied: Navigating fairness, bias, and ethics in educational ai applications](#). *Preprint*, arXiv:2407.18745.

Appendix A Example Source Item and User-Prompt

Part of the user-prompt for generating a multiple-choice item. The source item follows a strict HTML format embedded within JSON.

```
"english": {
  "options": [
    {
      "isCorrectAnswerKey": false,
      "text": "<html><head></head><body><p>91</p></body></html>"
    },
    {
      "isCorrectAnswerKey": false,
      "text": "<p>186</p>"
    },
    {
      "isCorrectAnswerKey": false,
      "text": "<p>1801</p>"
    },
    {
      "isCorrectAnswerKey": true,
      "text": "<p>1860</p>"
    }
  ],
  "prompt": "<p>There are 31 days in May. Calvin does 60 pushups every day of the month of May.</p><p><br/>How many total pushups does he do in the month of May?</p>"
},
"scoring": {
  "defaultScore": 0,
  "rubric": "D"
}
```

Now, generate 10 new, original items for grade 4 students that follow the same structure and style. The standards covered in these items is: 'Use the four operations with whole numbers to solve problems.' Follow the content specifications and fairness guidelines as constraints. Make the new items similar difficulty as the source item. Otherwise, be creative in the items you generate. The item content goes in the 'english' JSON field. Instructions on scoring the item goes in the 'scoring' JSON field. Return a JSON array structured similarly to the source.

Appendix B AI Reviewer Prompt

```
_REVIEW_SYSTEM = (  
    "You are a Smarter Balanced assessment content reviewer. "  
    "Evaluate generated items strictly and concisely. "  
    "Return only valid JSON - no markdown fences, no extra text."  
)  
  
_REVIEW_TEMPLATE = """  
Review this generated assessment item against the content specification.  
  
Content Specification:  
{content_spec}  
  
Fairness Guidelines:  
{fairness_guidelines}  
  
Generated Item:  
{item_json}  
  
Evaluate whether this item:  
1. Matches the target claim, target, and depth of knowledge (DOK)  
2. Has language and complexity appropriate for the grade level  
3. Has correct item structure (e.g., right number of answer options and correct answers)  
4. Avoids fairness or sensitivity issues  
  
Return ONLY this JSON object:  
{{"decision": "accept" | "flag" | "reject", "reasoning": "<1-2 sentences>"}}  
  
- "accept": item meets all requirements  
- "flag": minor issues that need human review before use  
- "reject": significant structural or content problems  
"""
```