CHAPTER 1

AUGMENTED INTELLIGENCE AND THE FUTURE OF ITEM DEVELOPMENT

Mark J. Gierl
University of Alberta

Hollis Lai
University of Alberta

Donna Matovinovic
ACT Inc.

Testing organizations require large numbers of diverse, high-quality, content-specific items to support their current test delivery and test design initiatives. But the demand for test items far exceeds the supply. Conventional item development is a manual process that is both time consuming and expensive because each item is written individually by a subject-matter expert (SME) and then reviewed, edited, and revised by groups of SMEs to ensure every item meets quality control standards. As a result, item development serves as a critical bottleneck in our current approach to content

Application of Artificial Intelligence to Assessment, pages 1–24

development for testing. One way to address this problem is to *augment* the conventional approach with computer algorithms to improve the efficiency and increase the scalability of the item development process. Automatic item generation (AIG) is the process of using models to produce items using computer technology. With AIG a single model can be used to produce hundreds of new test items. The purpose of our chapter is to describe and illustrate how augmented intelligence in item development can be achieved with the use of AIG. The chapter contains three sections. In the first section, we describe the conventional approach to item development. We also explain why this approach cannot be used to meet the growing demand for new test items. In the second section, we introduce augmented intelligence in item development and we describe how AIG can be used to support the human-machine interactions needed for efficient and scalable content production. In the third section, we provide a summary and we highlight directions for future research.

CONTEMPORARY ITEM DEVELOPMENT AND THE PROBLEM OF SCALABILITY

The conventional approach to item development is a manual process where SMEs use their experiences and expertise to produce new test items. It relies on a method where the SME creates each test item individually. Then, after each item is created, it is edited, reviewed, and revised until the item meets the required standards of quality (Haladyna & Rodriguez, 2013; Lane, Raymond, Haladyna, & Downing, 2016; Schmeiser & Welch, 2006). The SME is responsible for the entire process which involves identifying, organizing, and evaluating the content required for creating new items. This approach relies on human judgment acquired through training and experience. As a result, item development has often been described as an "art" because it depends on the knowledge, experience, and insight of the SME (Haladyna & Rodriguez, 2013; Schmeiser & Welch, 2006). Conventional item development is also a standardized process that requires iterative refinements to address quality control (Lane, Raymond, Haladyna, & Downing, 2016; Schmeiser & Welch, 2006). The item development process is standardized through the use of guidelines where SMEs are provided with information to structure their task in a consistent manner that produces reliable and valid test items (Haladyna & Downing, 1998; Haladayna, Dowing, & Rodriguez, 2002; Haladyna & Rodriguez, 2013). Standardization helps control for the potentially diverse outcomes that can be produced when different SMEs perform the same item development task. Guidelines provide a summary of best practices, common mistakes, and general expectations that help ensure the SMEs have a shared understanding of their tasks and responsibilities. Iterative refinement supports the practice of item development

through the use of a structured and systematic item review. That is, once an item has been written, it is then reviewed to evaluate whether it has met important outcomes described in the guidelines. Typically, reviews are conducted by committees of SMEs. Reviews can focus on a range of standards and objectives related to item content (e.g., does the item match the test specifications), fairness (e.g., does the item illicit construct-irrelevant variance due to subgroup differences), cognitive complexity (e.g., is the linguistic complexity of the item aligned to grade-level expectations), and presentation (e.g., is the item grammatically correct; Perie & Huff, 2016; Schmeiser & Welch, 2006). The review yields feedback on different standards of item quality that, in turn, can be used by the SME to revise and improve the original item.

The conventional approach has two noteworthy limitations. First, conventional item development is inefficient. It is both time consuming and expensive because it relies on the item as the unit of analysis (Drasgow, Luecht, & Bennett, 2006). That is, each item in the process is unique and therefore each item must be individually written, edited, reviewed, and revised. Many different components of item quality can be identified. For example, item quality can be determined, as noted in the previous paragraph, by the item content, fairness, cognitive complexity, and presentation. Because each item is unique, each component of item quality must be reviewed and, if necessary, each item must be revised. Because writing and reviewing is conducted by highly-qualified SMEs, the conventional approach is expensive.[1]

Second, conventional item development is challenging to scale in an economical way. The scalability of the conventional approach is again linked to the item as the unit of analysis. When one item is required, one item is written and reviewed by the SME. When 100 items are required, 100 items must be written and reviewed by the SMEs. Hence, a large number of SMEs who can write and review items is needed to scale the process. Conventional item development can result in an increase in item production when large numbers of SMEs are available. But item writing and reviewing is a time-consuming and expensive process due to the human effort needed to create, review, edit, and revise large numbers of new items.

These two limitations highlight the importance of establishing an efficient and scalable approach to item development. These limitations are also amplified in the modern era of educational assessment where test delivery and design are rapidly evolving to support different forms of on-demand testing. Test delivery marks the most important shift. Researchers and practitioners now recognize that educational testing is neither feasible nor desirable using the paper-based format. The cost of printing, scoring, and reporting paper-based tests requires tremendous time, effort, and expense. Computer-based testing (CBT) provides a viable alternative to paper-based testing that helps reduce delivery costs while providing important

benefits for examinees. CBT permits testing on-demand thereby allowing examinees to take the test at any time during instruction. Items on CBT are scored immediately thereby providing examinees with instant feedback. CBT allows for continuous administration thereby allowing examinees to have more choice about when they write their tests.

A second important change, made possible with the implementation of CBT, is the rapid expansion of test design. Many innovative test designs are now being used to create new types of computerized tests. For instance, computerized tests can be designed to identify examinee's cognitive problem-solving strengths and weaknesses (e.g., Hoijtink, Béland, & Vermeulen, 2014; van der Kleij, Eggen, Timmers, & Veldkamp, 2012; van der Kleij, Vermeulen, Schildkamp, & Eggen, 2015). Computerized tests can be paired with computer "agents" to engage examinees in conversational dialogues (Graesser, Li, & Forsyth, 2014). Computerized tests can be designed to score examinee's written and spoken responses and then to provide examinees with feedback on their responses using dynamic multimedia score reports (Shermis, Burstein, Brew, Higgins, & Zechner, 2016). These designs require computerized delivery. These designs must also be supported with large banks of items to permit frequent administration and to provide examinees with continuous feedback. A bank is a repository of items. Banks must be initially created with a set of items and then constantly replenished with new items to ensure that examinees receive a continuous supply of diverse, high-quality, content-specific items while limiting exposure to maintain test security. With a paper-based test designed to support summative test score inferences, a relatively small number of items was sufficient because the test was delivered in fixed length format across a small number of administrations in order to meet a specific purpose. Modern CBTs that incorporate innovative test design, by way of comparison, require large numbers of items because tests are delivered in variable length forms or with multiple forms at many times during the year in order to meet multiple purposes (e.g., examinees receive their summative score as well as detailed formative feedback on their problem-solving strengths and weaknesses). Unfortunately, the large banks required for modern educational testing are not readily available. Moreover, the means by which large numbers of items can quickly be developed using the conventional approach to satisfy these new banking requirements are unclear.

AUGMENTED INTELLIGENCE IN EDUCATIONAL TESTING

Artificial intelligence is a term for describing the theory and the application of computer systems that perform tasks and solve problems that normally require human intelligence (Russell & Norvig, 2010). Human-in-the-loop hybrid-augmented intelligence or augmented intelligence, for short is an

area within artificial intelligence that deals with how computer systems can emulate and extend human cognitive abilities thereby helping to improve human task performance and to enhance human problem-solving (Zheng et al., 2017). It requires the interaction between a computer system and a human in order for the system to produce an output or solution. Augmented intelligence combines the strength of modern computing using computational analysis and data storage with the human capacity for judgment to solve complex and, often, unstructured problems. Augmented intelligence can therefore be used to characterize any process or system that improves the human capacity for solving complex problems by relying on a partnership between a human and a machine (Pan, 2016).

While conventional item development relies entirely on the knowledge, experience, and insight of the SME, AIG is a process where models are first created by SMEs and then these models are used by a computer system to generate items (AIG; Gierl & Haladyna, 2013; Irvine & Kyllonen, 2002). Hence, AIG can be characterized as an augmented intelligence approach in educational testing because large numbers of new items can readily be manufactured using the coordinated inputs created by humans with outputs produced from computers. Gierl and Lai (2013, 2016a) described AIG as a three-step approach. First, the content for item generation is identified by the SME. This content is specified using principles and guidelines that highlight the knowledge, skills, and abilities required to solve problems and perform tasks in a specific content area. Second, an item model is developed by the SME to stipulate where the content must be placed to generate new items. Item models (LaDuca, Staples, Templeton, & Holzman, 1986) identify which parts of the assessment task can be manipulated for item generation. Third, computer algorithms place the content into the item model subject to rules. Assembly is conducted with algorithms because constraint programming is required to ensure that the content in the models is combined is a way that yields meaningful test items.

THE ROLE OF HUMAN EXPERTISE IN AIG

Content Definition

To begin, the content of interest must be identified for item development. Currently, the items for most educational tests are developed using test specifications. These specifications are created by SMEs, often, as a two-way matrix, where the rows represent content areas and/or learning outcomes and the columns represent cognitive skills (Haladyna & Rodriguez, 2013; Perie & Huff, 2016; Schmeiser & Welch, 2006). The content domain is organized by the structure of the topics and the content included within each topic. Mathematics, for example, is often structured in a hierarchy of

topics that range from simple to complex. The topics in mathematics also require the integration of simple concepts (e.g., simple addition and subtraction) to produce more complex concepts (e.g., application of addition and subtraction in factoring) as the topics increase in complexity. Alternatively, language arts is often structured using a list of key topics. But instead of developing the topics in a hierarchy like mathematics, language arts uses the same topic (e.g., vocabulary) but in different contexts (e.g., sentence structure). The topics in the content areas are then organized using descriptions of knowledge, skills, and abilities. The most widely used taxonomy for identifying these skills is Bloom's Taxonomy of Educational Objectives: Cognitive Domain (Bloom, Englehart, Furst, Hill, & Krathwohl, 1956) or some variation of Bloom's approach (e.g., Anderson & Krathwohl, 2001). The taxonomy ranges from the simplest level, knowledge (i.e., recall of specific information), to the most complex level, evaluation (i.e., the ability to judge the value of materials and methods for given purposes). Because SMEs have extensive knowledge and experience with the examinees, curriculum, and learning environment, they develop items for each cell in the test specifications by anticipating the cognitive skills that examinees will use to answer items correctly for topics in each content area.

A construct map can be used to identify the content required for item development. A construct map is a description of different knowledge and performance skills that are ordered from easy to complex on a score scale (Wilson, 2005). These skills are expressed as proficiency claims. The claims include statements about the knowledge and skills that examinees who score at different points on the scale should exhibit for a specific content area. Hence, the purpose of a construct map is to describe how a scale can be created that links test scores to test performance using specific claims about examinees' knowledge and skills in a content area. Construct maps can be used to guide item development because the proficiency claims include a description of the cognitive mechanisms within a content area for each task (Luecht, 2013; Mislevy, Steinberg, & Almond, 2003). Items are designed by SMEs to measure these claims.

The content for item development can also be identifying using cognitive models. A cognitive model specifies the thinking processes that underlie test performance (Leighton & Gierl, 2007). Gierl, Lai, and Turner (2012) described the development and use of a cognitive model, specifically, for AIG. This model highlights the knowledge, skills, and abilities required to solve a problem in a specific domain by organizing the cognitive- and content-specific information into a structured representation of how the SME expects examinees to think about and solve test items. The knowledge and skills specified in a cognitive model are identified using an inductive process by asking the SME to review an existing item and then to identify and describe information that could be used to create new items. A

cognitive model can be used to guide item development because it outlines the knowledge, skills, and abilities required to solve a problem in specific content areas. Items are designed by SMEs to measure these content-specific knowledge and skills.

Content Definition for AIG Reading Comprehension Example

To demonstrate how the content definition is used to guide item development, we provide an example from the domain of reading comprehension as measured on the ACT assessment. Reading comprehension on the ACT assessment requires examinees to

> derive meaning from several texts by (1) referring to what is explicitly stated and (2) reasoning to determine implicit meanings. Specifically, questions will ask you to use referring and reasoning skills to determine main ideas; locate and interpret significant details; understand sequences of events; make comparisons; comprehend cause-effect relationships; determine the meaning of context-dependent words, phrases, and statements; draw generalizations; and analyze the author's or narrator's voice and method. (ACT Inc., 2017)

An example of a section of a short prose passage from a released version of the ACT assessment is presented in Figure 1.1. After reading the passage, examinees are presented with a series of selected-response items about the

At Home in a Language

When I was a child growing up in Delhi, India. My parents and I will have spent our summers in Calcutta, India, visiting my grandparents, aunts, uncles, and cousins. We took the train over eight hundred miles from Delhi to Calcutta, which I considered a treat as itself. I loved the dining car, the cozy sleeping berth in our cabin, and the gentle rocking motion of the train that would lull me to sleep at night. As an adult, I prefer to travel by car. When we arrived at the Calcutta station the next morning, we were welcomed announcing train arrivals and departures over the intercom by the sound of the Bengali language.

Back in Delhi, the language most people commonly spoke was Hindi. Though I spoke Hindi fluently, it wasn't my first language. My parents were born in Calcutta, when most people spoke Bengali. They had lived there for many years before they go married and moved to Delhi, where Hindi was widely spoken. Because my parents had grown up speaking Bengali, we spoke Bengali, not Hindi, in our house. It was not surprising, then, that hearing Bengali on the streets of Calcutta made me feel right at home.

Figure 1.1 A section of a short prose passage taken from the released June 2011 administration of the ACT.

content in the passage. Each item is designed to measure one of three different skills: key ideas and details, craft and structure, and integration of knowledge and ideas (see ACT Inc., 2017, p. 7 for details). For the purpose of our chapter, the skill we will use to illustrate AIG is called "correcting inappropriate shifts in verb tense and aspect" which is found in the craft and structure category.

Item Model Design

After the content is identified, a test item that can be used to measure the content is created. Using a conventional approach, items are created individually to measure the content. Using AIG, the goal is to create many items simultaneously to measure the content. To achieve this goal, an item model for AIG must be designed. An item model specifies the components in a test item that can be manipulated to produce new items (Bejar et al., 2003; LaDuca et al., 1986). These components include the stem, options, and auxiliary information. The stem contains context, content, item, and/or the question the examinee is required to answer. The options include a set of alternative answers with one correct option and one or more incorrect options or distracters. Both stem and options are required for selected-response item models. Only the stem is created for constructed-response item models. Auxiliary information includes any additional content, in either the stem or option, required to generate an item, including text, images, tables, graphs, diagrams, audio, and/or video. Each component contain variables. Each variable, in turn, can have one or more integer and/or string values (i.e., values nested within variables for each component of the item model).

Often, item model design begins by identifying a parent item. The parent can be found by reviewing items from previous administrations or by drawing on a bank of existing test items. A decade ago, Gierl, Zhou, and Alves (2008) presented a list of 20 item models derived from existing parent items across a range of content areas that could be used for AIG. The parent item provides a starting point for item model design because it highlights the underlying structure for a potential model thereby providing a point-of-reference for creating alternative items. The statistical characteristics can also be used to identify well-functioning parents and, by implication, high-quality, content-specific test items. The SME creates a model using the parent by identifying variables within each component of the item that can be manipulated to produce new items. Hence, the parent serves as a template that can be used for item design. Alternatively, design can proceed by creating the item model from scratch without the use of a parent.

In addition to the model, the SME must also specify the values, which can be formatted as integer and/or string, for each variable in the model.

Item Model for AIG Reading Comprehension Example

An example of the parent item and the associated model derived from the short passage in Figure 1.1 is presented at the top of Figure 1.2. The item is: When I was a child growing up in Delhi, India. My parents and I will have spent our summers in Calcutta, India, visiting my grandparents, aunts, uncles, and cousins. The item contains the underlined phrase "I will have spent." The correct answer is "I would have spent." The model presented below the parent item in Figure 1.2 can be expressed using variables in the sentence as: [Variable 1, Part 1] [Variable 2, Part 1] [Variable 3] [Variable 1, Part 2] [Variable 4] [Variable 2, Part 2] [Variable 5]. The key feature in this model required to measure the content specification "correcting inappropriate shifts in verb tense" is that [Variable 1, Part 1] must contain the key determiner and [Variable 1, Part 2] must contain the cue. Our example also contains feasible alternative content strings for each variable in the model. For instance, the string options for the cue (i.e., Variable 1, Part 2) has six values: "spent," "will have spent," "spending," "will spend," "will be spending," and "have spent."

Summary

The SME is responsible for identifying, organizing, and evaluating the content and designing the models used for item generation. The content used for item development can be found in many different sources of information, including test specifications, construct maps, and cognitive models. These sources all provide different types of descriptions for the types of cognitive skills required to solve problems in specific content areas. Once the content is identified, item models are designed to measure the content. Item modeling is a method for scaling production by specifying the variables and the integer and/or string values that can be manipulated to produce new items. Because an item model serves as an expression of the SME's understanding of how examinees use their knowledge and skill to solve problems in a specific content area, many different item models can be created to capture this understanding. Hence, each model is distinct for each SME meaning different models can be produced by different SMEs. For this reason, defining content and designing models are considered to be creative tasks that rely on the knowledge and judgement of the SMEs acquired through their training and experience.

[Variable 1, Part 1][Variable 2, Part 1][Variable 3][Variable 1, Part 2][Variable 4][Variable 2, Part 2][Variable 5]

Variable 1, Part 1	Variable 2, Part 1	Variable 3		Variable 1, Part 2		Variable 4		Variable 2, Part 2		Variable 5
When I was a child growing up in	Delhi, India,	my parents	and	I would spend	our	summers	in	Calcutta, India,	visiting my	grandparents, aunts, uncles, and cousins.
During the year we lived in By the time I leave Were I to have stayed in	Buffalo, United States, Cologne, Germany, Tel Aviv, Israel,	my family my siblings my cousins		I spent I will have spent I would have spent I spending I will spend I will be spending I have spent NO CHANGE		weekends holidays vacations		Boston, United States, Berlin, Germany, Jerusalem, Israel,		grandparents aunts and uncles cousins

Figure 1.2 Item model and the string value for each variable in the model.

THE ROLE OF COMPUTER TECHNOLOGY IN AIG

The third step for item generation requires computer systems to place the content into the item model. Assembly is conducted with algorithms because it is a constraint logic programming task. Iterations assemble all possible combinations of the integer and/or string values for each of the variables, *subject to rules*. The rules are created using constraints applied to each integer and/or string value across all variables in the model. Without the use of constraints, all of the values in each of the variables would be iteratively combined to create new items. However, many of these combinations do not produce items that are sensible or useful. For instance, the city–country combination of [Variable 2, Part 1] in Figure 1.2 must be constrained so it corresponds with the appropriate city–country combination in [Variable 2, Part 2]. For example, Cologne, Germany must be matched with Berlin, Germany. It would not be reasonable to match Cologne, Germany with Boston, United States, particularly when [Variable 4] used the string "weekends." Constraints serve as restrictions that described the rules that must be applied during the assembly task so that meaningful items are generated. When the goal is to produce large banks containing many similar items, a relatively small number of constraints are applied to the variables in the item model. Alternatively, when the goal is to produce small banks containing different items, a relatively large number of constraints are applied to the variables in the item model. Hence, constraint coding is considered to be the source of "intelligence" in AIG. It is the method by which AIG captures the specific decisions, judgments, and rules used by SMEs to produce diverse, high-quality, content-specific test items. Constraint coding also yields *every* plausible combination of the values within the variables. As a result, both plausible content combinations that would be anticipated by the SME, and some that would possibly be unanticipated, are produced during generation.

The Logic of Constraint Coding for Item Generation

To generate items, conditions are expressed as constraints in the item model. Constraints initially expressed as Boolean logic conditions (see Gierl, Zhou, & Alves, 2008 for details) are now expressed as a binary, element-by-element, matrix to guide the generation process. The process of defining constraints begins by selecting a pair of variables in the item model. This selection results in the creation of a vector that specifies all element (i.e., integer and/or string value) combinations for the two selected variables. Next, constraints between the two variables for each value element are defined in binary, with 0 being non-permissible and 1 being permissible.

The constraints required for all variable pairing is then defined for every value element in the item model. Each constraint of a variable pair contributes a vector. Finally, all of the vectors from the constraint coding are combined to create a matrix that defines the permissible content combinations for the entire model. This matrix, in turn, guides the generation process by providing a detailed rule-based map where the product of each binary element define all permissible content combinations and where each permissible content combination is an individual test item. Large item models, which are common in operational AIG applications, often contain 8–12 integer and string values nested within 5–7 variables producing millions of content combinations, prior to the application of the constraints. Then, when the constraints are applied, the majority of the content combinations are eliminated because they produced infeasible combinations that would result in meaningless or inaccurate items. The combinations that remain are meaningful items produced by assembling specific combinations of integers and/or string values across the variables in the item model. Gierl, Lai, Hogan, and Matovinovic (2016) described a mathematics AIG example used for operational item development in a large testing company. They presented the results from 18 math item models that generated over 109,300 items. The largest model in their study contained seven variables. The string and/or integer values ranged from 3 to 12 across the seven variables in the model resulting in 1,270,080 generated items before the application of the constraint codes. The final bank for this model contained 11,145 items after the use of the constraints demonstrating that 99% of the content combinations were eliminated after the constraints were applied.

Item Generation with IGOR

Different types of computer systems have been written to generate test items (e.g., Gütl, Lankmayr, Weinhofer, & Höfler, 2011; Higgins, 2007; Higgins, Futagi, & Deane, 2005; Singley & Bennett, 2002). Gierl et al. (2008) created the IGOR computer system to conduct the assembly task. IGOR, which is the acronym for **I**tem **G**enerat**OR**, is a JAVA-based program that assembles test items using the integer and string values specified for each variable in the item model. That is, once the content is defined and the item model is created, IGOR systematically places the content into the model subject to the constraints to produce new items. To begin, the item model is entered in IGOR. The Item Model Editor window has a clearly structured interface that allows the user to input and structure each model. The editor has three panels corresponding to the components of either a selected- or constructed-response test item. The stem panel is where the stem for the item model is specified. The item model presented in Figure 1.2 is

presented in the stem panel in Figure 1.3. The variables panel is used to add and manipulate the integer and/or string values as well as to apply the constraints between variables. The options panel is used to specify the correct and incorrect alternatives. The options are classified as either a key or distractor. An example of constraint coding is provided in Figure 1.4. The city–country combination of [Variable 2, Part 1] in Figure 1.2 is constrained so it corresponds with the appropriate city–country combination in [Variable 2, Part 2]. Buffalo, United States is matched to Boston, United States for our reading comprehension AIG example. To generate items from a model, the Test Item Generator dialogue box is presented where the user specifies the size of the generated item bank and the location where the generated items will be saved. A full test bank with no IGOR coding can be generated or a portfolio which contains the item bank along with a list of the IGOR input can be produced. The user is also required to specify the

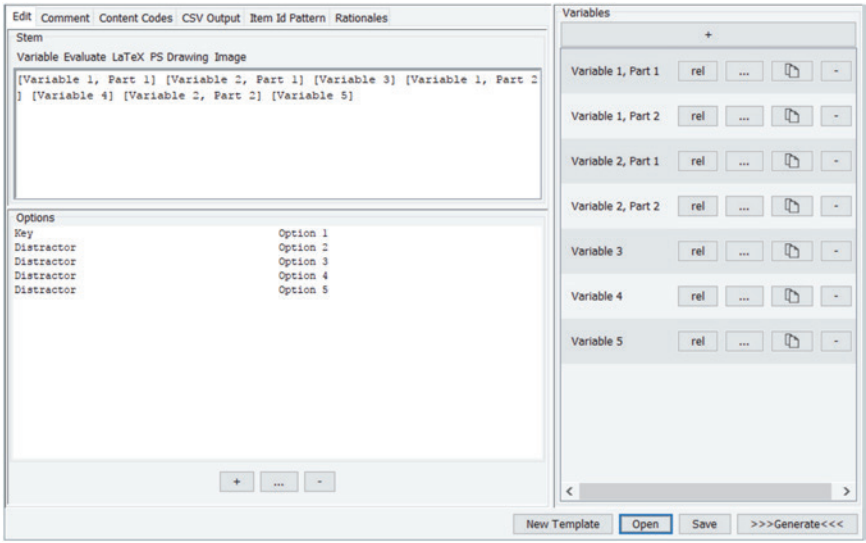

Figure 1.3 The Item Model Editor window in IGOR.

Figure 1.4 An example of a constraint in IGOR.

generation options. These options include the size of the generated item bank, the order of the options, the number of options for each generated item, and the format (i.e., with or without key; raw or formatted).

Generation Using the Reading Comprehension Item Model

In our example, constraint coding is required to generate the stem, the correct option, and the distractors for the item model in Figure 1.2. Four different sets of constraints are defined. The first set of constraints specify the relationship between [Variable 1, Part 1], which contains the key determiner, and [Variable 1, Part 2], which contains the cue, for the correct option. This constraint is used to align the appropriate key determiner and cue to produce the correct option. The second set of constraints define how the content for [Variable 1, Part 2] will be used to produce the distractors. Each multiple-choice item contains one correct option and four distractors, where one of the distractors will always be the option "NO CHANGE." Hence, three of the seven distractors will be selected with the constraint that each distractor for each item is unique for every generated item. One of the options must include "NO CHANGE." The third set of constraints define the relationship between [Variable 2, Part 1] and [Variable 2, Part 2], which both contain locations. In our example, the country variables India, the United States, and Germany for Part 1 will be matched with Part 2 for [Variable 2] so the same country is used in Parts 1 and 2 (see Figure 1.4). The fourth set of constraints describe the relationship between [Variable 3] and [Variable 5], which both contain family members and/or relatives. In our example, cousins is used for both variables. As a result, a constraint is specified so that cousins cannot be used for both [Variable 3] and [Variable 5] in the same generated test item. Using the item model in Figure 1.2 without constraints produced 5,832 items. After the constraints were applied, 83 items were generated for the reading comprehension example.

Summary

To generate test items, constraints are specified across each variable in an item model using the Item Model Editor in IGOR (see Figure 1.5). These constraints are stored as an explanatory matrix that contain detailed rules for combining content. This matrix is used to identify all feasible combinations of integer and/or string values for every variable and these combinations, in turn, are assembled to produce a bank of generated items. Large

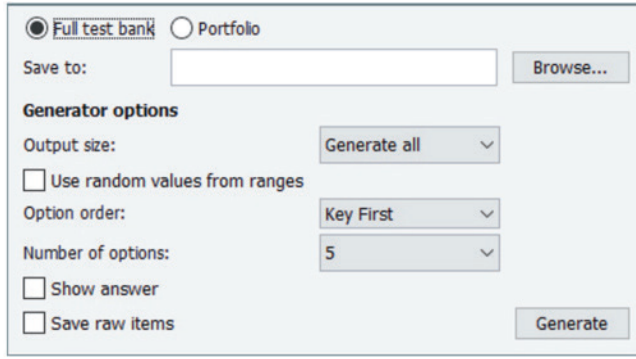

Figure 1.5 The Test Item Generator dialogue box in IGOR.

item models, initially producing millions of content combinations, are reduced to a relatively small number of meaningful integer and/or string combinations after the application of the constraints. The generation process on a modern high-speed computer requires only seconds to complete.

SUMMARY AND DIRECTIONS FOR FUTURE RESEARCH

In their 2006 chapter on "Test Development" in the 4th edition of the handbook *Educational Measurement*, Cynthia Schmeiser and Catherine Welch begin with this provocative question—Test Development: art or science? When augmented intelligence is applied to educational testing using AIG the answer is art *and* science. AIG is a process where the content required for testing is first identified by the SME and then the model required to process this content is created by the SME. Content specification and item model design require human expertise, experience, and judgment. These models, in turn, are used to generate large numbers of high-quality, content-specific test items using the algorithmic power of modern computing and the logic of constraint programming. Computational power, constraint coding, and data storage are features of a computer system. By merging the outcomes from the content-based creative task with the technology-based generative task, automated processes can be used to create a new approach to item development. AIG, therefore, represents a merger between the art and science of item development where well-defined responsibilities that adhere to specialized skills and the appropriate task assignments contribute to the production of diverse, high-quality, content-specific items using an approach that is both efficient and scalable.

Benefits of Item Generation Over Conventional Item Development

AIG can be used to directly address the two limitations inherent to conventional item development. First, AIG is an efficient approach for content creation because it relies on the item model, rather than the item, as the unit of analysis. AIG treats the item model as the fundamental unit of analysis where a single model is used to generate many items compared with the conventional approach where the item is treated as the unit of analysis. The item model serves as a structured template where the SME is responsible for manipulating only specific, well-defined, elements in the model. The remaining elements are not altered during the AIG process. Because the item model can be used to yield many test items compared with developing each item individually, quality is easier to monitor and maintain. The view of an item model as a template with both fixed and variable content contrasts with the conventional view of a single item where every element is unique, both within and across items.

Second, AIG is a process that is easily scaled, unlike conventional item development. AIG permits the SME to create a single item model that, in turn, yields many test items. Hence, AIG is scalable because one item model can generate many test items. When one item is required, one item model is written by the SME. But when 40 items are required, one item model may be sufficient, depending on the content specification. As a result, the number of required items is no longer tied to the number of SMEs who can write and review items. Rather, production is linked to the number of available item models, where a small number of SMEs can create the models that produce large numbers of new items. Because of this unit of analysis shift, the cost per item will decrease because SMEs are producing item models that yield multiple items rather than producing single unique items. As a result, building banks using SMEs who implement AIG could be an efficient and sustainable approach to item development.

DIRECTIONS FOR FUTURE RESEARCH

Scaling Item Production

Expanding Output Within and Between Item Models

Once the content has been identified and the item model has been structured, as outlined in Figure 1.2, the content for each of the variables can easily be expanded in order to increase the generative capacity for each model. For instance, new phrases can be added for [Variable 1, Part 1], new cities and countries can be added for [Variable 2, Part 1] and [Variable 2, Part 2],

and new family and extended family members can be added [Variable 3] and [Variable 5]. An expanded list of relationships between the key determiner and the cue can also be specified. New key determiners and cues can be identified within an existing sentence or in new sentences in the passage.

In the model presented in our example, relationships between cities and countries, immediate and extended family members, and tenses for verbs was used to generate items. These value and variables combinations could be used to create a database library of cities and countries, family members, and verb tenses. Subsequent models that require one or more of these variables could use the information in the library to generate content. For example, once the cities and countries database is complete, that list could be used for *any* model that includes a cities and country variable. Or, new models could be designed that intentionally include a cities and country variable to make use of the generative capacity that exists when using content in the database. These types of AIG content libraries do not currently exist.

Implementing n-Layer Item Modeling

The goal of item generation using the item model is to produce new test items by manipulating the values within the variables using rules garnered from the constraints. The example in our chapter is described as a 1-layer item model because a relatively small number of values and variables were used. But a limitation of using a 1-layer item model is that relatively few variables can be manipulated because the number of potential manipulations is fixed to the total number of words in the stem. As an alterative, *n*-layer item models can be created (Gierl & Lai, 2012). The goal of AIG using the *n*-layer item model is to produce items by manipulating the variables at two or more levels in the model. Unlike the 1-layer model where the manipulations are constrained to a linear set of generative operations using the variables at a single level, the *n*-layer model permits manipulations of a nonlinear set of generative operations using variables at multiple levels. As a result, the generative capacity of the *n*-layer model is comparatively high. The concept of *n*-layer item generation is based on a hierarchical structure of content meaning that values are often embedded within one another. This hierarchical organization can also be used as a guiding principle to generate large numbers of meaningful test items. The use of an *n*-layer item model is therefore a flexible template for expressing different structures thereby permitting the development of many different but feasible combinations of embedded variables. The *n*-layer structure can be described as a model with multiple layers of variables, where each variable can be varied simultaneously at different levels to produce different items.

A comparison of the 1-layer and *n*-layer item model is presented in Figure 1.6. For this example, the 1-layer model can provide a maximum of four different values for Variable A. Conversely, the *n*-layer model can provide

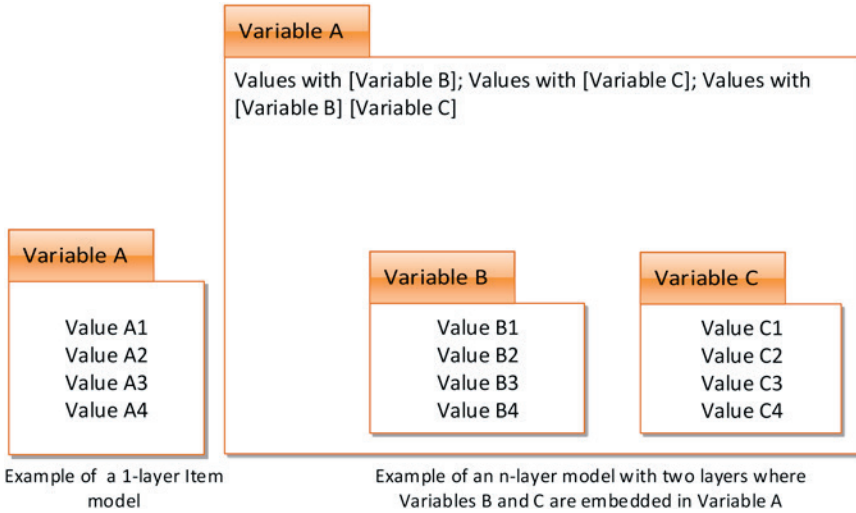

Figure 1.6 A comparison between a 1- and *n*-layer item model.

up to 64 different values by embedding the same four values for Variables B and C within Variable A. Because the maximum generative capacity of an item model is the product of the ranges in each element (Lai, Gierl, & Alves, 2010), the use of an *n*-layer item model will always increase the number of items that can be generated relative to a 1-layer structure. To date, the application of *n*-layer item modeling has been limited (for an example, see Gierl et al., 2015). However, the potential for *n*-layering to increase the generative capacity of an item model is significant.

Refining and Expanding the AIG Process

Quality Control

The SME is responsible for creating the content and the logic specified in the item model. This logic should be evaluated by another SME as part of the AIG quality control process necessary for any operational item development system. Using conventional item development, SMEs evaluate new items individually using guidelines and standards for quality. But with AIG, the item model is a structured template where the SME is responsible for manipulating specific, well-defined, elements in the model. The remaining elements are not altered during the AIG process. As a result, costly yet common errors in item development (e.g., including or excluding words, phrases, or expressions along with spelling, grammatical, punctuation, capitalization, typeface, and formatting problems) can be avoided because

only specific elements in the stem and options are manipulated across a large number of items. In other words, the item model serves as a template for which the SME manipulates only a small number of clearly-defined elements. The remaining elements are not altered during development.

One other important consequence of creating item models is that the unit of analysis in quality control moves from the item to the model level. The SME is responsible for identifying the content and specifying the logic for each item model. If the model is correctly specified, then the generated items will reflect the correct combination of the content and logic. Hence, quality control is focused on the more manageable task of evaluating the quality of a small number of AIG models rather than reviewing large numbers of individual test items. Gierl and Lai (2016b) described how the content and the logic specified in the item models could be evaluated using a standardized rating scale. The outcome of the rating would, in turn, serve as feedback for improving the model thereby ensuring the generated items meet high standards of quality. The effectiveness of using rating scales to improve the quality of the models and the generated items has not yet been evaluated.

Creating Item Development Ecosystems

Educational testing is now in a constant and unprecedented state of change. As noted earlier, significant change is occurring in how educational tests are delivered and designed. In addition to delivery and design, educators are also beginning to change the purpose and the longstanding conventions associated with educational testing. In 2015, ACT Inc. released a report called "Beyond Academics: A Holistic Framework for Enhancing Education and Workplace Success" (Camara, O'Connor, Mattern, & Hanson, 2015). The authors claim that accountability systems of the past have focused exclusively on measures of academic achievement when attempting to understand college and career readiness. This limited construct definition, they claim, is far too narrow for understanding the complex factors that affect the school to work transition for today's students. To expand the construct, ACT developed a holistic framework that includes a comprehensive list of the knowledge, skills, behaviors, and competencies across four areas related to college and work readiness as well as a description of how these skills change as students progress from kindergarten to career:

> Development of this the framework is based on a comprehensive review of relevant theory, education and work standards, empirical research, input from experts in the field, and a variety of other sources for each of four broad domains [core academic skills; cross-cutting capabilities; behavioral skills; education and career navigation skills]. The framework spans the K–Career continuum, since the precursors of success emerge very early in life and development continues well beyond the confines of traditional secondary and

> postsecondary education. To describe what people need to know and be able to do across this continuum, both the education and work readiness framework and the associated model of success are defined in terms of critical transitions. These transitions differ slightly across the four broad domains, but some major transitions are relevant to all four . . . For each of these transitions, the framework describes what individuals need to know and be able to do to be successful. A hierarchical taxonomy within each broad domain organizes the more specific dimensions and the knowledge and skills and provides a common language for describing the precursors of success. The focus is ultimately on knowledge, behaviors, and skills because these are amenable to change. Thus, the taxonomy is well positioned to inform education and other interventions aimed at helping people achieve education and workplace success. Because the taxonomy is holistic and comprehensive, it can be used to identify new ways to assess and improve education and workplace readiness. (Mattern & Hanson, 2015, p. 9)

This description demonstrates how educators are beginning to radically change the scope and purpose of their tests. AIG is an example of how augmented intelligence can be used to increase the efficiency and scalability of item production for any testing scope. But when the purpose of the test is also questioned, a more complex problem must be addressed: How do we create large numbers of new test items in an efficient and cost-effective manner to measure complex, dynamic, and, potentially, ill-structured tasks characteristic of new and evolving constructs such as college and career readiness?

One way to address this problem is by drawing on the benefits of a *problem-solving ecosystem*. Crowdsourcing is one example of a problem-solving ecosystem. In this ecosystem, large complex tasks are decomposed into smaller more simple tasks. Participants drawn from a broad range of content areas and backgrounds then contribute to solving the tasks. The tasks are embedded in a workflow which ensures that the participants in each step of the process use and augment solutions provided by participants from the previous step. This workflow, in turn, is embedded in the problem-solving ecosystem which is designed to systematically address a larger more complex multi-faceted problem.[2]

In an *item development ecosystem*, the cognitive capabilities of the participants could be used to create diverse item models for measuring complex task performance in a unique construct such as "studying and learning" (ACT cross-cutting capabilities; O'Connor, Gambrell, & Pulvermacher, 2015) or "maintaining composure" (ACT behavioral framework; Casillas, Way, & Burrus, 2015). These cognitive outputs, in turn, could then be supplied as inputs for an AIG computer system that uses the information from the models to produce large numbers of new items to measure these novel constructs. The ecosystem requires an online workspace where participants

can connect, contribute, combine, revise, evaluate, and integrate their ideas and data using a common framework (Michelucci & Dickenson, 2016). The workspace also includes a well-designed user interface so participants can easily and readily interact with the AIG computer system. An item development ecosystem must include strategies for scaling the item development process and for monitoring the quality of the generated content. But most of all, the ecosystem needs a broad range of human participants who have diverse backgrounds and experiences. Recall that the "art" of item modeling yields an expression of the SME's understanding of how students use their knowledge and skill to solve problems in a specific content area. But different item models can be created from many diverse perspectives to capture this understanding. In this chapter, we demonstrated how this understanding is expressed in a well-defined content area like English language arts. Presumably, it can also be expressed in an ill-defined content area like "studying and learning" and "maintaining composure." Each item model is distinct for each participant meaning that different but equally valid models can be produced by different participants in the ecosystem. Some of the models will be characteristic of the SME perspective. This perspective is predictable and expected because SMEs are content experts who have item development experience. But the item development ecosystem is not just composed of SMEs. Other models will be characteristic of a different and possibly unanticipated perspective. For instance, the item models designed by a developmental psychologist to capture critical education and workplace transitions may differ from the models created by the English language arts SME. Finally, interactions between participants (e.g., the developmental psychologist and the English language arts SME) may lead to yet a different outcome. In sum, problem-solving ecosystems where the unique contextual knowledge and rich cognitive capabilities of humans interact seamlessly with powerful computing systems to produce both predictable and novel outcomes may provide the most promising environment for solving the complex item development challenges that await researchers and practitioners in educational testing. An item development ecosystem has yet to emerge. But the conceptual foundation, the item development methods, and computational tools now exist to make this type of ecosystem possible.

NOTES

1. Rudner (2010) estimated that the cost of developing *one operational item* for a large-scale test using the conventional item development approach ranged from $1,500 to $2,500 USD (p. 157).
2. In our view, creating large numbers of new test items to measure many evolving concepts and constructs related to college and career readiness that begin in kindergarten is an example of a complex multi-faceted problem.

REFERENCES

ACT Inc. (2017). *Preparing for the ACT test.* Iowa City, IA: Author.

Anderson, L. W., & Krathwohl, D. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives.* New York, NY: Longman.

Bejar, I. I., Lawless, R., Morley, M. E., Wagner, M. E., Bennett, R. E., & Revuelta, J. (2003). A feasibility study of on-the-fly item generation in adaptive testing. *Journal of Technology, Learning, and Assessment, 2*(3). Retrieved from https://ejournals.bc.edu/index.php/jtla/article/view/1663

Bloom, B., Englehart, M., Furst, E., Hill, W., & Krathwohl, D. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain.* New York, NY: Longman.

Camara, W., O'Connor, R., Mattern, K., & Hanson, M. (2015). *Beyond academics: A holistic framework for enhancing education and workplace success* (ACT Research Report Series Vol. 4). Iowa City, IA: ACT.

Casillas, A., Way, J., & Burrus, J. (2015). Behavioral skills. In W. Camara, R. O'Connor, K. Mattern, & M. Hanson (Eds.), *Beyond academics: A holistic framework for enhancing education and workplace success* (pp. 25–38). Iowa City, IA: ACT.

Drasgow, F., Luecht, R. M., & Bennett, R. (2006). Technology and testing. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 471–516). Washington, DC: American Council on Education.

Gierl, M. J., & Haladyna, T. (2013). *Automatic item generation: Theory and practice.* New York, NY: Routledge.

Gierl, M. J., & Lai, H. (2016a). Automatic item generation. In S. Lane, M. Raymond, & T. Haladyna (Eds.), *Handbook of test development* (2nd edition; pp. 410–429). New York, NY: Routledge.

Gierl, M. J., & Lai, H. (2016b). A process for reviewing and evaluating generated test items. *Educational Measurement: Issues and Practice, 35*, 6–20.

Gierl, M. J., & Lai, H. (2012). Using automatic item generation to create items for medical licensure exams. In K. Becker (Chair), *Beyond essay scoring: Test development through natural language processing.* Paper presented at the annual meeting of the National Council on Measurement in Education, Vancouver, BC.

Gierl, M. J., & Lai, H. (2013). Using automated processes to generate test items. *Educational Measurement: Issues and Practice, 32*, 36–50.

Gierl, M. J., Lai, H., Hogan, J., & Matovinovic, D. (2015). A method for generating test items that are aligned to the Common Core State Standards. *Journal of Applied Testing Technology, 16,* 1—18.

Gierl, M. J., Lai, H., & Turner, S. (2012). Using automatic item generation to create multiple-choice items for assessments in medical education. *Medical Education, 46,* 757–765.

Gierl, M. J., Zhou, J., & Alves, C. (2008). Developing a taxonomy of item model types to promote assessment engineering. *Journal of Technology, Learning, and Assessment, 7*(2). Retrieved from https://ejournals.bc.edu/index.php/jtla/article/view/1629

Graesser, A., Li, H. I., & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science, 23*, 374–380.

Gütl, C., Lankmayr, K., Weinhofer, J., & Höfler, M. (2011). Enhanced Automatic Question Creator – EAQC: Concept, development and evaluation of an automatic test item creation tool to foster modern e-education. *Electronic Journal of e-Learning, 9,* 23–38.

Haladyna, T. M., & Downing, S. M. (1989). Validity of a taxonomy of multiple-choice item-writing rules. *Applied Measurement in Education, 2,* 37–50.

Haladyna, T. M., Downing, S. M., & Rodriguez, M. C. (2002). A review of multiple-choice item-writing guidelines for classroom assessment. *Applied Measurement in Education, 15*, 309–333.

Haladyna, T. M., & Rodriguez, M. C. (2013). *Developing and validating test ttems.* New York, NY: Routledge.

Hoijtink, H., Béland, S., & Vermeulen, J. (2014). Cognitive diagnostic assessment via Bayesian evaluation of informative diagnostic hypotheses. *Psychological Methods, 19,* 21–38.

Higgins, D. (2007). *Item distiller: Text retrieval for computer-assisted test item creation* (RM-07-05). Princeton, NJ: Educational Testing Service.

Higgins, D., Futagi, Y, & Deane, P. (2005). *Multilingual generalization of the Model Creator software for math item generation* (RR-05-02). Princeton, NJ: Educational Testing Service.

Irvine, S. H., & Kyllonen, P. C. (2002). *Item generation for test development.* Hillsdale, NJ: Erlbaum.

LaDuca, A., Staples, W. I., Templeton, B., & Holzman, G. B. (1986). Item modeling procedures for constructing content-equivalent multiple-choice questions. *Medical Education, 20,* 53–56.

Lai, H., Gierl. M. J., & Alves, C. (2010, April). Using item templates and automated item generation principles for assessment engineering. In R. M. Luecht (Chair), *Application of assessment engineering to multidimensional diagnostic testing in an educational setting.* Paper presented at the annual meeting of the National Council on Measurement in Education, Denver, CO.

Lane, S., Raymond, M., Haladyna, R., & Downing, S. (2016). Test development process. In S. Lane, M. Raymond, & T. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 3–18). New York, NY: Routledge.

Leighton, J. P., & Gierl, M. J. (2007). Defining and evaluating models of cognition used in educational measurement to make inferences about examinees' thinking processes. *Educational Measurement: Issues and Practice, 26,* 3–16.

Luecht, R. M. (2013). An introduction to assessment engineering for automatic item generation. In M. Gierl & T. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 59–101). New York, NY: Routledge.

Mattern, K., & Hanson, M. (2015). ACT holistic framework of education and work readiness. In W. Camara, Camara, R. O'Connor, K. Mattern, & M. Hanson (Eds.), *Beyond academics: A holistic framework for enhancing education and workplace success.* (pp. 1–9). Iowa City, IA: ACT.

Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1*, 3–67.

Michelucci, P., & Dickinson, J. (2016). The power of crowds. *Science, 351,* 32–33.

O'Connor, R., Gambrell, J., & Pulvermacher, R. (2015). Cross-cutting capabilities. In W. Camara, Camara, R. O'Connor, K. Mattern, & M. Hanson (Eds.), *Beyond academics: A holistic framework for enhancing education and workplace success.* (pp. 19–24). Iowa City, IA: ACT.

Pan, Y. (2016). Heading toward artificial intelligence 2.0. *Engineering, 2,* 409–413.

Perie, M., & Huff, K. (2016). Determining content and cognitive demands for achievement tests. In S. Lane, M. Raymond, & T. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 119–143). New York, NY: Routledge.

Rudner, L. (2010). Implementing the graduate management admission test computerized adaptive test. In W. van der Linden & C. Glas (Eds.), *Elements of adaptive testing* (p. 151–165), New York, NY: Springer.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Schmeiser, C. B., & Welch, C. J. (2006). Test development. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 307–353). Westport, CT: National Council on Measurement in Education and American Council on Education.

Shermis, M., Burstein, J., Brew, C., Higgins, D., & Zechner, K. (2016). Recent innovations in machine scoring of student- and test taker-written and spoken responses. In S. Lane, M. Raymond, & T. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 335–254). New York, NY: Routledge.

Singley, M. K., & Bennett, R. E. (2002). Item generation and beyond: Applications of schema theory to mathematics assessment. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development* (pp. 361–384). Mahwah, NJ: Erlbaum.

van der Kleij, F. M., Vermeulen, J. A., Schildkamp, K., & Eggen, T. J. (2015). Integrating data-based decision making: Assessment for learning and diagnostic testing in formative assessment. *Assessment in Education: Principles, Policy & Practice*, *22*(3), 324–343.

van der Kleij, F. M., Eggen, T. J., Timmers, C. F., & Veldkamp, B. P. (2012). Effects of feedback in a computer-based assessment for learning. *Computers & Education*, *58*, 263–272.

Wilson, M. (2005). *Constructing measures: An item response modelling approach.* Mahwah, NJ: Erlbaum.

Zheng, N., Liu, Z., Ren, P., Ma, Y., Chen, S., Yu, S., . . . Wang, F. (2017). Hybrid-augmented intelligence: Collaboration and cognition. *Frontiers of Information Technology & Electronic Engineering, 18,* 153–179.