Graduate
Management
Admission
Council®

Using Automatic Item Generation to Address Item Demands for CAT

Hollis Lai, Cecilia Alves, and Mark J. Gierl

University of Alberta

Presented at the CAT Research and Applications Around the World Poster Session, June 2, 2009

Abstract

Item writing is a craft that is both resource intensive and time consuming. In order to minimize item exposure, computer adaptive testing requires a large number of test items for administration. As test development become more complex, demand for the quantity and quality of items far exceeds the capacity of item writers writing in the current fashion. To address the demand for items to be created in a precise and high-volume fashion, this study suggests the use of automatic item generation (AIG) as a solution to current test development practices. The purpose of this paper is to explore development preferences of item models under AIG, using an item model taxonomy. After adequate training, 34 item models were created and categorized under the taxonomy, and generated 64,280 items. Implications from the study and potential challenges for applying AIG are discussed.

Acknowledgment

Presentation of this paper at the 2009 Conference on Computerized Adaptive Testing was supported in part with funds from GMAC®.

Citation

Lai, H., Alves, C., & Gierl, M. J. (2009). Using automatic item generation to address item demands for CAT. In D. J. Weiss (Ed.), ***Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing.*** **Retrieved [date] from www.psych.umn.edu/psylabs/CATCentral/**

Author Contact

Hollis Lai, CRAME, 6-110 Education North, University of Alberta, Edmonton, Alberta, Canada, T6G 2G5. Email: hollis.lai@ualberta.ca

Using Automatic Item Generation to Address Item Demands for CAT

Computerized adaptive testing (CAT) are currently making tremendous demands on item banks for three reasons. First, as test length increases in CATs to provide adequate content coverage, requirements for the number of test items increase to ensure test scores are reliable (Wainer & Eignor, 2000). Second, a focus on cognitive psychology and evidence-centered test designs has prompted the need for more items to measure a much more specific set of skills. Thus, more test items have to be created, but they would only be used for specific cases. Third, item exposure concerns have required items to be administered at a minimal rate to ensure item security. The issues presented have a common solution, and that is to provide a large number of high quality test items. The purpose of this paper is to (1) demonstrate the development of items under automated item generation (AIG), and (2) explore what type of item models are created under an item model taxonomy (Gierl, Zhou & Alves, 2008). A study generating 64,280 items under 34 item models is presented to determine if there is a preference for item model types when created by test developers.

Background of AIG

The concept of AIG has been around for nearly 30 years (Irvine & Kyllonen, 2002) but has only gathered attention for development recently. AIG involves creating test items in an automated manner under a predefined structure known as an item model. There are numerous approaches for generating items using a computer (Millman & Westman, 1989), but they generally require the availability of an item model. An item model (Bejar, 2002; Drasgow et al., 2006) is a general prototypical representation of the items to be generated. For example, a multiple-choice item model consists of three components. The *stem* is the portion of the item that provides context for the item, and poses a question or problem that examinees will be required to answer. The *option*s are a set of alternatives presented with the correct answer to allow the examinee to choose the correct response. *Auxiliary information* is the additional material that supplements the item by illustrating the stem and/or the options using tables, images, or diagrams. Furthermore, each component of an item model can contain both fixed and variable elements (Gierl, Zhou & Alves, 2008). Currently, multiple-choice item models are the most widely used and easy to implement models; while other types of item models are available, they are comparatively less popular.

The AIG Process

Computationally, the current approach to implementing AIG is straightforward. Generating items requires an item model, and iteratively modifies the template's elements through its range in all possible combinations to generate new items. The item model can be further constrained such that (1) illogical or unfitting items will not be generated, (2) there are greater differences in the interval of iterations to minimize similar items, and (3) only certain combination of variable elements will be generated if a sequence of response is required. Test developers have to specify all components and constraints of an item model, and the item generator will generate all possible new items constrained by the item model.

In contrast, the logical considerations for developing an item model are comparatively more complex. The original purpose of AIG was developed to satisfy the content specifications and psychometric demands of a test. This was soon discovered to be insufficient, as cognitive specification and models were required for item models (Gitomer & Bennett, 2003). In order to express the specifications of the item model, and to ensure that the generated items are representing the intent of the item model, an item model can be developed from two general approaches (Drasgow et al., 2006). The first approach is to create item models under a theory of difficulty that would predetermine the psychometric properties of its generated items. This is known as generating items from a *strong theory* (Irvine, 2002). With this approach, all generated items are within an expected range of psychometric properties, as the level of item difficulty is estimated by making assumptions about the cognitive processes needed to solve the item. While this approach is associated with a high upfront cost, as psychometric models of skills have to be blueprinted for the purposes of constraining psychometric properties, strong theory might be beneficial as it might not require field testing for calibration. Moreover, *strong theory* is also suited for generating items under an existing strong cognitive theory (Drasgow et al., 2006), where items can be written in a systematic manner to fit the students' level of cognitive complexity.

Since the *strong theory* approach can be resource intensive if there is no existing psychometric model to reference item difficulty, an alternative approach to generate items from a broader domain is to generate from *weak theory*. Under this approach, the item models are created under a design framework that stipulates the types of features in an item model that affect difficulty and those that do not. This design guideline can be generalizable across a broad trait, as it could incorporate many features that would apply in the domain of content (Gitomer & Bennett, 2003). Using this guideline and pre-calibrated items across a range of difficulty, item models are created to generate items within the expected level of difficulty by varying features similar to the pre-calibrated items of the nearest level of difficulty.

Item Model Taxonomy

Development of item models has revealed that, depending on the level of difficulty and the content area of the model, there might be preferred levels of variation in the item model. In order to delineate the types of variations within an item model, the models were categorized in order to better comprehend how models vary. To identify the possible item model types, Gierl, Zhou, and Alves (2008) developed a taxonomy to categorize and delineate the levels of variation in components of the parent item model. The item model taxonomy (Gierl et al., 2008) suggested that item models can be categorized into two types of variations. First, item stems can vary in four ways. An item stem can be presented with:

1. Independently varied components.
2. Dependently varied components—components might vary depending on the state of another component
3. Mixture of dependent and independent components.
4. Fixed component—the stem will not be varied.

Second, options of an item model can be classified into three types of variation:

1. Random—options are randomly selected to be used.
2. Constrained—options are select based on other varied components, such as the stem.
3. Fixed—the same set of options are presented for all generated items

From the different combinations of stem and option types, eleven item model types can be conceptualized (excluding fixed stem with fixed options, see Figure 1). As little is known about each item model type and its feasibility for item development, this study explored the preferred model types for item generation. Furthermore, the item model taxonomy has been used to create exemplars of each item model type, but it has not been applied in a test development environment. By identifying preferred item model types, the results may be used to guide and ease model development for specific item types.

Figure 1. The Eleven Types of Item Models By Type of Variation

	Stem			
Options	Independent	Dependent	Mixed	Fixed
Randomly Selected				
Constrained				
Fixed				

Similar to traditional item writing, generated items also have to meet expected psychometric properties. Traditional methods of item writing are known for having deviations from the expected levels of difficulty during calibration (Drasgow et al., 2006). Although there has yet to be a study investigating the accuracy of expected psychometric properties in generated items, the use of guidelines and pre-calibrated items to guide item model writing is expected to allow generated items to outperform traditional items with their expected levels of difficulty.

Rationale for Implementing AIG

The overarching reason for implementing AIG is obvious; it has the ability to solve current test development issues in an efficient manner. But more specifically, there are three distinct benefits that make AIG a logical development for large-scale CAT programs.

Cost Benefits

Test item writing is simply too costly for test development in the 21st century. Not only are test developers limited in the number of items they can write in one day, but once an item is created it might not be used for other test development reasons. Testing organizations such as Educational Testing Service estimate that 10% of the total testing cost is accounted for by item writing (Wainer, 2002). Moreover, the price of writing new items increases dramatically once the costs of editing, field-testing, and calibrations are taken into account. Furthermore, the price to develop conceptually difficult or abstract reasoning based tests are higher, as difficult items tend to be more time consuming to develop than easier items (Wainer, 2002). Although there are no exact figures on how much AIG will save compared to traditional item writing, it is widely acknowledged that item model development is more effective than traditional item writing (Wainer, 2000). Consider the benefits of AIG that if item models can be developed at the same pace as writing an item, where item models can generate multiple items, the number of generated items from AIG would eclipse the number of written items exponentially. Although, the cost implication of using AIG has yet to be documented, as there has yet to be a cost comparison study of AIG, but its potential benefits are promising.

Enhancing Test Security and Decreasing Item Exposure

Since CAT usually draw items as a function of an examinee's ability estimated from his or her responses, more items would be needed across the entire range of difficulty. To avoid examinees of a similar ability level receiving the same questions, different items would need to be drawn at the same ability. AIG can help minimize this process of item exposure as it can generate many comparable items at the same level of difficulty. Moreover, test security breaches often occur when large-scale assessments reuse test items in different administrations (Wainer, 2000). The use of AIG could enhance test security and minimize item exposure if used with CAT. AIG could improve test security by introducing different variants of the same item to ensure that different items are administered. While some common items generated from the same item model should not appear on the same form, the availability of AIG would decrease item exposure because similar items could be drawn from the item bank (Gitomer & Bennett, 2003). Item exposure is measured by the number of times an item was used compared to the total number of test administrations.

More Accurate Estimate of Examinee Abilities

With the emergence of diagnostic testing and CAT, large-scale assessments have started to focus on improving the accuracy of inferences of examinees' ability (Thompson et al., 2003). AIG is well suited to developing items for innovative test designs, where AIG would enable modern testing methods such as diagnostic testing and CAT to make more accurate estimates of examinee abilities. AIG can contribute to diagnostic testing by generating items from item models that target the measurement of specific weaknesses in the examinee's ability. This added information is beneficial, as questions that probe the skill can be administered more than once to improve reliability. When the entire test is constructed from the use of item models that probe different weaknesses, the test would be able to make a more accurate estimate of the examinee's ability compared to individually crafted items. Similarly, CAT could benefit from AIG by generating more items across the entire range of difficulty, thereby allowing the test to include more questions that focus in on the examinee's level of ability.

As CAT designs evolve to incorporate more complex test designs, it is apparent that more items are needed. AIG becomes an essential tool to enable complex CAT designs for large-scale administration.

Method

In addition to demonstrating the AIG process, the introduction of the AIG process also allowed us to explore the model development processes. Specifically, the purpose of this study was to (1) develop and categorize item models under the taxonomy (Gierl et al., 2008) to explore characteristics of model types for specific content areas and level of difficulty, and (2) determine which, if any, item model types are favored for AIG development by developers.

Sample

In order to explore the item generation process, this study introduced the AIG process to a group of 12 test developers in a provincial testing program. The test developers were responsible for writing items for four content areas (Math, Science, Language Arts, and Social Sciences) across three grades (Grades 3, 6, and 9). Furthermore, each test developer had a minimum of two years of teaching experience and had a minimum of one year experience in writing test items.

Procedure

The deployment of AIG was completed in three phases. First, the test developers were trained on AIG. Over a period of six weeks, a weekly two-hour session was spent on training the test developers. A range of topics was covered during the sessions, including: introduction to AIG, how to create item models, the types of item models as described in the taxonomy, and tutorials on developing item models. Second, developers were encouraged to create item models for item writing. This took place in a four-week period, where developers were free to create item models as they saw fit, during which weekly consultation sessions were available for developers to consult with the authors on their approaches to item modeling. Third, after item models were created, items were generated from the models. In order to generate items from item models, an item generator named IGOR (see Gierl et al., 2008, for a description of IGOR) was used in this project. The number of generated items was tabulated and the item models were categorized within the taxonomy. Semi-structured interviews were also completed with the developers at the end of the study to inquire about their experience with AIG.

Results

After training, test developers created 34 item models across four subject areas fitting within their test specifications. While each developer was responsible for more than one content area and grade level, not all test developers created models for item generation, as the use of AIG was voluntary. Table 1 contains the number of models created for their corresponding content area and grade level, and the Appendix includes a sample of the item models.

From the 34 item models created, IGOR was able to generate 64,280 items (as shown in Table 1). Although the number of generated items was impressive, the generated items were not the focus of this study. The focus of this study was to explore how the item models fit under the model taxonomy. As the developers were trained in the types of variation that exist, they were not told which type of models to create but were encouraged to develop models from all types of variations that were most convenient to them. After the models were finalized by the developers, the models were categorized into the taxonomy. Table 2 show the respective number of models created under the categorization of the taxonomy.

Table 1. Number of Item Models and Number of Items Generated By Content Area and Grade Level

Content Area	Grade Level	Number of Item Models	Number of Items Generated
Language Arts	3	5	7,709
	6	2	2,160
Mathematics	3	6	11,782
	9	11	41,409
Science	3	1	26
	9	8	456
Social Studies	6	1	718
Total		34	64,260

From the results in Table 2, four points can be noted from the distribution of item models in the taxonomy. First, only seven of the 11 model types in the taxonomy were employed in this study, with the developers preferring fixed options least. Second, a model type that is most popular with developers was independently varied stems with constrained options—20 of the 34 item models in this study were of this type. Third, developers of language arts and social studies items had more trouble creating item models, as they created nine of the 34 models in the same time frame as their science and math counterparts, where the models only accounted for 16.5% of the total number of generated items. Fourth, developers of language arts and social studies preferred to use fixed stems in their item model, where six of the nine models contained a fixed stem. These findings suggest that developers might have preferences for model types, and are likely to create item models with independently varied stems and constrained options.

Table 2. Number of Item Models Created Under the Item Model Taxonomy

	Stem			
Options	Independent	Dependent	Mixed	Fixed
Randomly Selected	None	1 Math model 1 Science model	None	2 Language Arts model 1 Social Studies model
Constrained	15 Math models 3 Science models 1 Social Studies models 1 Language Arts model	None	1 Math model 1 Language Arts model 1 Science model	3 Language Arts model
Fixed	1 Science model	1 Science model	None	Not Applicable

Discussion and Conclusions

The purpose of this study was to develop item models for AIG and explore what type of item models are the preferred types under the item model taxonomy. Although the present study was not able to show statistical differences, due to an insufficient number of item models created, the information presented in Table 2 showed that it is likely that independent and constrained model types are preferred over other types, from the test developer's perspective. Associated with its limitations, this study also demonstrated some deployment issues with AIG, specifically with introducing the idea of item model development to test developers in a testing situation.

Challenges in Implementing AIG

One main challenge for this study was the lack of item models created to detect any statistically significant trends of item model use. This reason for the lack of models was in part due to some deployment issues with AIG. First was an issue with implementing training for test developers. Although writing test items in a traditional manner is considered an art form, developing item models for AIG is conceptually different. All developers were willing to learn and dedicated effort to developing item models; however, item models often required background knowledge in computer programming in order to logically express the item model to effectively generate items. This requirement is problematic as it frustrated most test developers. Furthermore, the use of item models changed the scope of the finished product, where item writing would only be concerned with the quality of one item, developing an item model has to ensure that all generated items must meet the expected item quality. As a result of this technological hurdle, some test developers were frustrated with the development process and had gone back to item writing.

Another challenge encountered was with developing models for different content areas. Recognized in other studies (Fletcher, 2003), the method to generate items currently favors content areas that involve computational variations (such as science and mathematics) compared with content domains that are more verbal based (such as language arts and social sciences). This bias in difficulty for developing item models was the likely cause for the low number of models from the verbal-based content areas. Furthermore, as developers for math became more accustomed with developing item models, their models increased in complexity, such as the inclusion of numerous model constrains and variable elements, whereas item models from the verbal areas remained at the same level of complexity. To investigate this difference between content areas, Educational Testing Service has invested greatly in developing an AIG method that would be flexible for all content areas (Singley & Bennett, 2002; Wainer, 2002), but so far they have also only demonstrated a viable method of AIG for math. Studies have explored the use of science content in item modeling, but there is still a lack of development for an AIG method that can be used easily across all platforms.

Because of the limitations developers had encountered, they saw AIG as a supplementing tool, where not all items can be created effectively by item models. In addition, the test developers also acknowledged that item models might be the best approach in some cases, but they also agreed that item model development was currently more fitting for mathematics and some science items. This study and its limitations outlined above provide a glimpse of the potential challenges that might appear if AIG was to be implemented. Although there might seem to be uncertainty in implementing AIG, it is an item development method that can address the needs of current test designs. AIG can populate item banks and allow effective CAT administration. Moreover, since the risks associated with implementing AIG are low and indirect from the public, the resources required to implement AIG are not demanding, and the potential benefits of AIG are quite high.

Conclusion

We demonstrated an implementation of AIG and categorized the resulting models in the item model taxonomy to learn the practical preferences of item model types. Our results were limited by the low number of item models developed but did provide some insights about test developers' preferences for item model types. In addition, this study will also contribute to the literature of AIG, as few implementations have been documented (Irvine, 2002). Although this study was conducted in a relatively small context, the large number of generated items from the small set of item models demonstrates the potential of AIG to create large item banks for CAT. Also, for reasons of test security and item exposure, demand for items is known to rise to ensure that students are answering questions that have not been previously administered (Bartley, 2006). Implementing AIG can enable both objectives by providing large amounts of test items. This, in turn, focuses future development of AIG on improving the quality of generated test items. Future studies include developing item models under assessment engineering and evidence centered test designs.

References

Bartley, J. (2006). Assessment is as assessment does: A conceptual framework for understanding online assessment and measurement. In M. Hricko & S. Howell (Eds.), *Online Assessment and Measurement: Foundations and Challenges.* Hershey, PA: ISP.

Bejar, I. (2002). Generative testing: From conception to implementation. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development*. Hillsdale, NJ: Erlbaum. p.199-217.

Bejar, I., Lawless, R., Morley, M., Wagner, M., Bennett, R. & Revuelta, J., (2002) A feasibility study of on-the-fly item generation in adaptive testing. *GRE Board Report 98-12.* Princeton, NJ: Educational Testing Serviceq.

Bugbee, A. (1996). The equivalence of paper and pencil and computer-based testing. *Journal of Research on Computing in Education*, 28(3), 282-290.

Drasgow, F., Luecht, R. & Bennett, R. (2006). Technology and testing. In R. Brennan (Ed.) *Educational Measurement 4th Ed.* Wesport,CT:NCME/ACE.

Duesbery, L., Ketterlin-Geller, L., McCoy, J. & Tindal, G. (2006). A computer-adaptive mathematics test accommodates third grade students with special needs in the Pacific Northwest. In M. Howell & S. Hricko (Eds.) *Online Assessment and Measurement: Case studies from higher education K-12 and corporate.* Hershey, PA: Idea-Group. p.157-169.

Educational Testing Service. (1994). *Computer-based tests: Can they be fair to everyone?* Princeton, NJ: ETS.

Ferrara, S. & DeMauro, G. (2006). Standardized assessment of individual achievement in K-12. In Brennan, R. (Ed.) Educational Measurement, 4th Ed. Wesport, CT:NCME/ACE.

Fletcher, J. (2003). Is it worth it? Some comments on research and technology in assessment and instruction. in *Technology and Assessment: Thinking Ahead: Proceedings of a Workshop.* Washington, DC: Board on Testing and Assessment, Center for Education.

Gitomer, D. & Bennett, R. (2003). Unmasking constructs through new technology, measurement theory and cognitive sciences, in *Technology and Assessment: Thinking Ahead: Proceedings of a Workshop.* Washington, DC: Board on Testing and Assessment, Center for Education.

Irvine, S. (2002). The foundations of item generation for mass testing. In S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development*. Hillsdale, NJ: Erlbaum. pp.3-32.

Larkin, K. & Weiss, D. (1975) *An empirical comparison of two-stage and pyramidal adaptive ability testing.* Minneapolis: University of Minnesota.

Linn, D. (2002). Using electronic assessment to measure student performance. *Educational Policy Studies Division: National Governors Association.* Retrieved April, 2009, from World Wide Web: http://www.nga.org/cda/files/ELECTORNICASSESSMENT.pdf

Millman, J., & Westman, R. (1989). Computer-assisted writing of achievement test items: toward a future technology. *Journal of Educational Measurement,* 26(2), 177-190.

Singley, M. & Bennett, R. (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*. Hillsdale, NJ: Erlbaum. pp.361-384.

Thompson, S., Thurlow, M., & Moore, M. (2003). *Using computer-based tests with students with disabilities* (Policy Directions No. 15). Minneapolis, MN: University of Minnesota, National Center on Educational Outcomes. Retrieved April 1, 2009, from the World Wide Web: http://education.umn.edu/NCEO/OnlinePubs/Policy15.htm

Van der Linden, W. & Glas, C. (2001). *Computerized testing: Adaptation and practice.* Boston, MA: Kluwer.

Wainer, H. (2002). On the automation generation of test items: Some whens, whys and hows in S. H. Irvine & P. C. Kyllonen (Eds.), *Item generation for test development*. Hillsdale, NJ: Erlbaum, pp. 287-305.

Appendix. Samples of the Item Models

Model #13:	**Stem:** ***Independent*****; Options:** ***Constrained*****; Auxiliary Information:** ***None***
	Item Model Variables
Stem	If the expression -I1x + I4 + I2x - I5 + I3x - I6 is simplified, which of the following response is a simplified version of the expression?
Elements	I1 Value Range: 2 – 5 by 1 I2 Value Range: 2 – 5 by 1 I3 Value Range: 2 – 5 by 1 I4 Value Range: 6 – 9 by 1 I5 Value Range: 5 – 8 by 1 I6 Value Range: 5– 9 by 1
Options	A. (-I2 - I3 + I1)x + I5 - I4 + I6 B. (-I2 - I3 + I1) x + I4 - I5 - I6 C. (I2 + I3 - I1)x + I5 - I4 + I6 D. (I2 + I3 - I1)x + I4 – I5 – I6
Auxiliary Information	None
Key	D

Model #21:	**Stem:** ***Independent*****; Options:** ***Fixed*****; Auxiliary Information:** ***None***
	Item Model Variables
Stem	The water quality of S1 and S2 located near S3 that have been clear-cut is affected because:
Elements	S1 Range: "river", "lake", "water ways", "estuaries" S2 Range: "streams", "creeks", "reservoirs", "tributaries", "other bodies of water" S3 Range: "forest regions", "woodland", "trees", "forests", "fields"
Options	A. nutrient availability decreases B. animal population decrease C. soil erosion increases D. solar heat increases
Auxiliary Information	None

Model #22:	**Stem:** ***Dependent*****; Options:** ***Randomly Selected*****; Auxiliary Information:** ***None***
	Item Model Variables
Stem	Ling uses number tiles to make a pattern. I1 I2 I1 ? ? ? I1 I3 I1 I2 I1 I3 The missing numbers in the above pattern are
Elements	I1 Value Range: 1 – 9 by 1 I2 Value Range: 0 – 9 by 1 I3 Value Range: 0 – 9 by 1 I1 ≠ I2 ≠ I3
Options	Key: I3 I1 I2 Distractors: I1 I3 I2; I3 I2 I1; I3 I1 I3; I3 I2 I3; I1 I3 I1; I1 I2 I1
Auxiliary Information	None
Key	C

Model #24: **Stem:** ***Dependent*****; Options:** ***Fixed*****; Auxiliary Information:** ***None***

	Item Model Variables
Stem	The elements S1 and S2 are both
Elements	S1 Range: “lithium”, “beryllium”, “sodium”, “magnesium” S2 Range: “boron”, “carbon”, “nitrogen”, “oxygen”, “fluorine”, “neon”, “aluminum”, “silicon”, “phosphorous”, “sulfur”, “chlorine”, “argon” As S1 = “lithium” or “beryllium”, S2 = “boron”, “carbon”, “nitrogen”, “oxygen”, “fluorine”, or “neon” As S1 = “sodium” or “magnesium”, S2 = “aluminum”, “silicon”, “phosphorous”, “sulfur”, “chlorine”, or “argon”
Options	A. in the same group B. non-metals C. metals D. in the same period
Auxiliary Information	None
Key	D

Model #27: **Stem: *Mixed*; Options: *Constrained*; Auxiliary Information: *None***

	Item Model Variables
Stem	A S1 has a power rating of I1 watts. If the S1 is left on for I2*60 minutes, then the amount of energy used, to the nearest joule, is ________ J.
Elements	I1 Value Range: 30, 40, 50, 250, 300, 1000, 1100, 1250, 1550 I2 Value Range: 2 – 6 by 0.5 S1 Range: “lightbulb”, “toaster”, “toaster oven”, “blender”, “television”, “room air condition”, “blow dryer”, “CD player”, “ceiling fan”, “coffee maker” As I1 = 40, S1 = “lightbulb”, As I1 = 1100, S1 = “toaster”, As I1 = 1550, S1 = “toaster oven”, As I1 = 300, S1 = “blender”, As I1 = 250, S1 = “television” As I1 = 1000, S1 = “room air condition”, As I1 = 1250, S1 = “blow dryer”, As I1 = 30, S1 = “CD player”, As I1 = 50, S1 = “ceiling fan, As I1 = 1000, S1 = “coffee maker
Options	A. I1 * I2 * 30 B. I1 C. I1 * I2 * 60 D. I1 * I2
Auxiliary Information	None
Key	D

Model #30:	Stem: *Fixed*; Options: *Randomly Selected*; Auxiliary Information: *None*
	Item Model Variables
Stem	Which of the following goals would **best** replace the question mark in Source IV?
Options	<u>Key:</u> “Reduce the number of women who receive prison sentences” “Increase public awareness and promote the abolition of incarceration of women” “Increase cooperation among women’s groups to address poverty, racism, and discrimination” <u>Distractors:</u> “Relocate female prisoners to rehabilitation facilities” “Increase the number of women who remain in custody” “Transfer female prisoners to jails that are closer to their families” “Increase the funding for constructing women’ prisons” “Increase the efficiency of dealing with women in the judicial system” “Increase the number of female judges” “Reduce the bail conditions for women accused of criminal offences”
Auxiliary Information	None
Key	C

Model #33: Stem: *Fixed*; Options: *Constrained*; Auxiliary Information: *Novel excerpt*

	Item Model Variables
Stem	Which of the following quotations is an example of onomatopoeia?
Options	<u>Key:</u> "bikes thundered over the bridge" (lines 5 – 6) "she hit the water with a crash" (line 20) "amid great whoops and splashes" (lines 16 – 17) "She came up spluttering" (line 21) <u>Distractors:</u> "his trusty steed leading the way" (line 2) "when Jack abruptly changed course" (line 4) "the races had been forgotten" (lines 1 – 2) "Slowly Charles walked his bike towards Jack" (line 8) "It's best to be standing up for that part" (line 11) "make sure you let go so the bike falls first" (line 12) "Faster and faster he went" (line 24) "he pulled his bike out of the lagoon" (line 25) "there were scratches all over the frame" (line 26) <u>Constraints:</u> If one of the four options begins with a capital letter, at least one other distractor needs to start with a capital letter. All distractors and key should be similar in length, if possible. In the case of the quotes from the passages listed on the left, they need to be listed in the order that they appear in the text when they are used as distractor/key.
Auxiliary Information	Novel excerpt
Key	C