Automatic Item Text Generation in Educational Assessment

Cédrick Fairon (1), David M. Williamson (2)

(1) Centre de traitement électronique des documents (CETEDOC)
University of Louvain (UCL)
1348 Louvain-la-Neuve, Belgium
fairon@tedm.ucl.ac.be

(2) Educational Testing Service (ETS)
Rosedale Road, Princeton, NJ - USA
dmwilliamson@ets.org

Abstract

We present an automatic text generation system (ATG) developed for the generation of natural language text for automatically produced test items. This ATG has been developed to work with an automatic item generation system for analytical reasoning items for use in tests with high-stakes outcomes (such as college admissions decisions). As such, the development and implementation of this ATG is couched in the context and goals of automated item generation for educational assessment.

Keywords – Mots Clés

NLG, ATG, CAT, INTEX, Finite State Transducers.

Génération automatique de texte, Tests informatisés, INTEX, Transducteurs à états finis.

1 Introduction

There is a significant research effort at Educational Testing Service (ETS) directed at the development of techniques for the automatic generation of test items in various types of assessment, including mathematics, verbal, and analytical reasoning applications. The potential benefits of such automatic item generation capabilities are substantial. The field of testing for high-stakes decisions places significant demands on the item generation capabilities of even large corporations. The potential for test items to be memorized and distributed to other examinees limits the number of times an item can be administered before it must be 'retired' and no longer used. When the test is a large-scale nationwide examination, such as the SAT or the GRE[1], with thousands of examinees per year this

[1] The SAT (test required for entering at the University) and the GRE (test required for starting graduate studies) are two national standardized tests: see http://www.ets.org/tests.html.

exposure limitation demands that there be a constant production line of new items to replace items that have been administered many times.

The high number of items that must be continuously 'retired' and replaced in large scale assessment creates a great need for a continuous item supply. At the same time, the rigorous review and pretesting processes that items are subjected to typically result in substantial time delay between writing and administering items. Therefore, automated item generation systems that are capable of automatically producing items, ensuring that the language of such items is appropriate, and providing an accurate prediction of the statistical characteristics of the item is of significant value to test development.

We will be presenting an ATG system aimed at creating 'ready to use' items for analytical reasoning tests (as used in the GRE). We will also show how the linguistic data and templates used in the generation process are distinct from the generation algorithms. This separation of data and templates from the source code facilitates maintenance and updating of the system as well as the adaptability to multiple languages (e.g. to date, this potential has been explored with English and French).

2 Analytical Reasoning Items

An analytical reasoning item (see Figure 1 for a human-written AR item retired from the GRE)[2] is typically composed of several elements. The initial element is the *scenario*, the situation description, that serves as part of the *context*, which "contextualizes" the exercise. The first paragraph (plain text) in Figure 1 is the initial scenario. As part of this scenario the number of *objects*, the elements to be manipulated (e.g. radio programs), and *slots*, the placement opportunities for the objects (e.g. radio program sequence), are specified. The context also includes a set of *restriction rules* that place constraints on the relationships in the context. In Figure 1, the restriction rules are provided in italics. The scenario and restriction rules jointly comprise the context. The actual AR *item stem* refers to the context and poses a particular question the examinee must solve[3]. In Figure 1 the AR item stem, which introduces an additional restriction rule, follows the context and is underlined. Immediately following the item stem is a series of *options* (the choices in a multiple choice item), which are bulleted in Figure 1. One of these options is the *key* (the third option in Figure 1), a correct response to the item stem and the only option which satisfies all the given rules, while the remaining options are incorrect, or *distracters*. The examinee must use their analytical reasoning skills to consider all restriction rules and select the key. The item stem and the options, in combination, represent the AR *item*. There are typically multiple items that refer to a single context which, taken together constitute an item set. There are usually numerous such AR sets in an assessment that uses these items.

[2] Figure 1 item is a particular subtype of AR item called *linear ordering*.

[3] For AR items there are three types of questions to appear in item stems: to indicate which of the following options *can* be true, *must* be true or *cannot* be true, in accordance with the given restriction rules.

A popular radio show has exactly six segments--film review, interview, news, political essay, song, and weather--that are aired consecutively, one after another, from first through sixth, without intervening material. Each segment is aired exactly once and in a manner consistent with the following restrictions:

The song must be aired earlier than the interview.
The news and the song must be aired consecutively.
The weather and the film review must be aired consecutively.
The interview cannot be aired consecutively with the weather.
The interview cannot be the next segment aired after the song.

If the political essay is the next segment aired after the song, which one of the following can be true?

- The film review is the fourth segment aired.
- The film review is the sixth segment aired.
- The interview is the fourth segment aired.
- The interview is the fifth segment aired.
- The weather is the fifth segment aired.

Figure 1: Example of an Analytical Reasoning Item

It is apparent that the syntax and the lexicon used are simple and rather repetitive, even to the point of appearing peculiar if taken out of context. This is the result of adherence to strict item-writing processes for AR items and while the result provides a poor example of naturalistic writing, it adheres to a fixed format for AR items in assessment and provides a standardized item format that facilitates examinee understanding and performance on multiple instances of such items. As such, this adherence to fixed structure helps to control the item difficulty by restricting the language structure.

3 Goal and needs

The goal of our ATG system is, for a given *scenario*, to generate the natural language for *restriction rules* to complete the *context* and then generate the required number of natural language items (item stems and options) to complete the item set. The scenarios themselves are not automatically generated but are stored in a digital database library and, upon generation of the abstract structure (see section 3.2) for the restriction rules, item stems and options, are accessed to determine the appropriate scenario to select and 'clothe' the generated abstract representations.

In the terminology of computer-based testing (CBT) automated item generation offers a new approach to *item banking*[4]. Using automated item generation the creation of test forms is not restricted to selection of items from a pre-existing item database but instead, for any *scenario* contained in the item bank, the system is able to generate hundreds of items, each with known statistical characteristics based on generation principles. In fact, this approach is equivalent to having an item bank containing all possible items, each with known statistical properties.

From the NLG point of view, our system is a hybrid system including Templates (Reiter, 1995) and aspects of NLG systems.

[4] "Item banking covers any procedures that are used to create pilot, analyze, store, manage and select test items so that multiple test forms can be created from subsets of the total "bank" of items." (Brown, 1997 p. 44).

3.1 Abstract representation of the item

A collaboration between Plymouth University and ETS has produced a system that automatically generates the fundamental abstract representation of AR items (Dennis, Handley, Bradon, Evans, & Newstead, 2002). Specifically, by establishing abstract representations of the number of objects and slots, the initialization rules, the stem rule, and the options. An example of this representation of an item, transformed into XML encoding for communication with the ATG, can be found as Figure 2 corresponding to the same item presented in Figure 1. This system was completed prior to the work on the ATG for "clothing" these abstract representations in natural language.

```
<ITEM OBJ=6 SLOTS=6>
<STIMULUS>
        <RULE>aboBE</RULE>
        <RULE>adjCE</RULE>
        <RULE>adjFA</RULE>
        <RULE>NadjBF</RULE>
        <RULE>NimaBE</RULE>
</STIMULUS>
<ITEM type="possibility">
        <RULE>imaDE</RULE>
<OPTIONS>
        <RULE>latA4</RULE>
        <RULE>latA6</RULE>
        <RULE>latB4</RULE>
        <RULE>latB5</RULE>
        <RULE>latF5</RULE>
</OPTIONS>
</ITEM>
```

Figure 2. Formal representation of an item

Each of the rules is formed with an *operator* (e.g. *abo, adj, lat, ima*)[5] and a couple of *objects* (the elements to be ordered). Rules can be coordinated (e.g. exclusive disjunction, symbolized by X, conjunction, symbolized by A, etc.). For example: *X aboCA aboAF* expresses an exclusive disjunction of two rules: *aboCA* and *aboAF*. Rules can also be modified with a *not* operator (N). By combining rules and variables, the abstract representations developed by Plymouth University can generate a very large number of valid formal relationships to be expressed in items.

3.2 Scenario data bank

As previously indicated, the *scenarios* used in the 'clothing' process are stored in a database. This library would contain a large number of hand-written scenarios, formatted in XML, and manually encoded series of lexical units for each scenario (to be used in the templates). Future research will address the development of a component that automates the selection of lexical information compatible with the scenario.

```
<CONTEXT>A popular radio show has exactly six segments--film review,
interview, news, political essay, song, and weather--that are aired
consecutively, one after another, from first through sixth, without
intervening material.  Each segment is aired exactly once and in a manner
consistent with the following restrictions:
</CONTEXT>
<LING>
   <ABO>earlier than</ABO>
   <ADJ>consecutively</ADJ>
   <N>segment,segments</N>
   <V>aired</V
</LING>
<ELEMENTS>
   <ELMT>the film review</ELMT>
   <ELMT>the interview</ELMT>
   <ELMT>the news</ELMT>
   <ELMT>the political essay</ELMT>
   <ELMT>the song</ELMT>
   <ELMT>the weather</ELMT>
</ELEMENTS>
```

Figure 3. Formal representation of an item

[5] *aboAB* means A is before B, *latB3* means B is the third element, *imaAB* means B immediately follows A, *adjAB* means that A and B are adjacent

4 Wording process

The wording process of the items is based on three elements: an abstract representation of the objects and rules (provided by the Plymouth system), some lexical information (related to the scenario and hand-coded in the database) and a finite-state grammar that describes, for each type of rule, some possible syntactic structures. The wording process is conducted as follows:

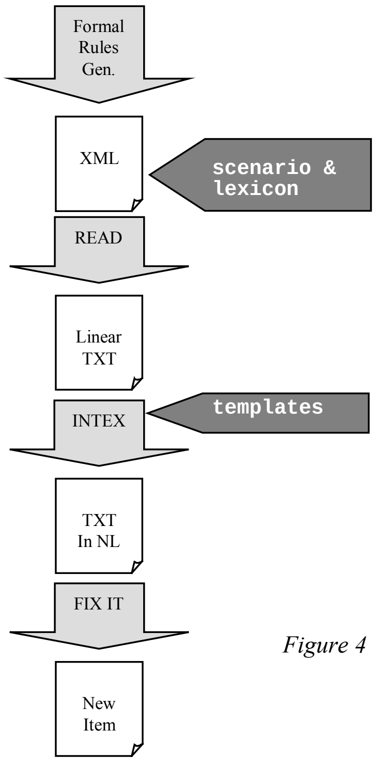
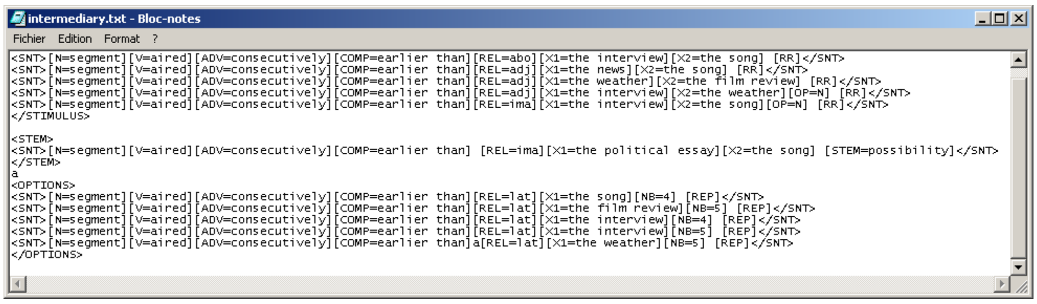


Figure 4

1) the formal representation of the item is combined with the context and its lexical data in a single XML document that will be fed to the system ('XML' in Fig. 4). Basically, the resulting XML document is a combination of the texts presented in Fig. 2 and 3;
2) a XML parser ('READ' in Fig. 4) reads the document, combines the rules (a step called *factorization*, performed in accordance with the difficulty model) and produces a linear output (one line for each natural language sentence to produce, see Fig. 5). At this point, each of these lines bear all the lexical elements that will be needed to complete the templates (an example of this output is provided as Fig. 5);
3) the template (a finite state transducer represented as an INTEX[6] graph, see Fig. 6) is applied on the intermediary text representation and generates the sentences in natural language. That *realization process* is performed with a generic INTEX program that applies the transducer on the text;
4) a polishing program ('Fix It' object in Fig. 4) fixes minor linguistic problems (case, agreement, elisions, etc.) by applying local rules.



```
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=abo][X1=the interview][X2=the song] [RR]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=adj][X1=the news][X2=the song] [RR]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=adj][X1=the weather][X2=the film review] [RR]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=adj][X1=the interview][X2=the weather][OP=N] [RR]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=ima][X1=the interview][X2=the song][OP=N] [RR]</SNT>
</STIMULUS>

<STEM>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than] [REL=ima][X1=the political essay][X2=the song] [STEM=possibility]</SNT>
</STEM>
a
<OPTIONS>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=lat][X1=the song][NB=4] [REP]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=lat][X1=the film review][NB=5] [REP]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=lat][X1=the interview][NB=4] [REP]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than][REL=lat][X1=the interview][NB=5] [REP]</SNT>
<SNT>[N=segment][V=aired][ADV=consecutively][COMP=earlier than]a[REL=lat][X1=the weather][NB=5] [REP]</SNT>
</OPTIONS>
```

Figure 5. Intermediary representation of the sentences

4.1 "Finite State templates"

The templates have been created under the form of Finite State Graphs with the INTEX graph editor (Silberztein, 1993). Figure 6 shows a simplified view of one "template-graph".

[6] INTEX is a corpora parser based that has been developed by Max Silberztein. It uses Finite State Automata to represent large coverage linguistic resources (http://www.nyu.edu/pages/linguistics/intex/). For a presentation of various INTEX applications, see Fairon (1998; 1999) and Silberztein (2000).

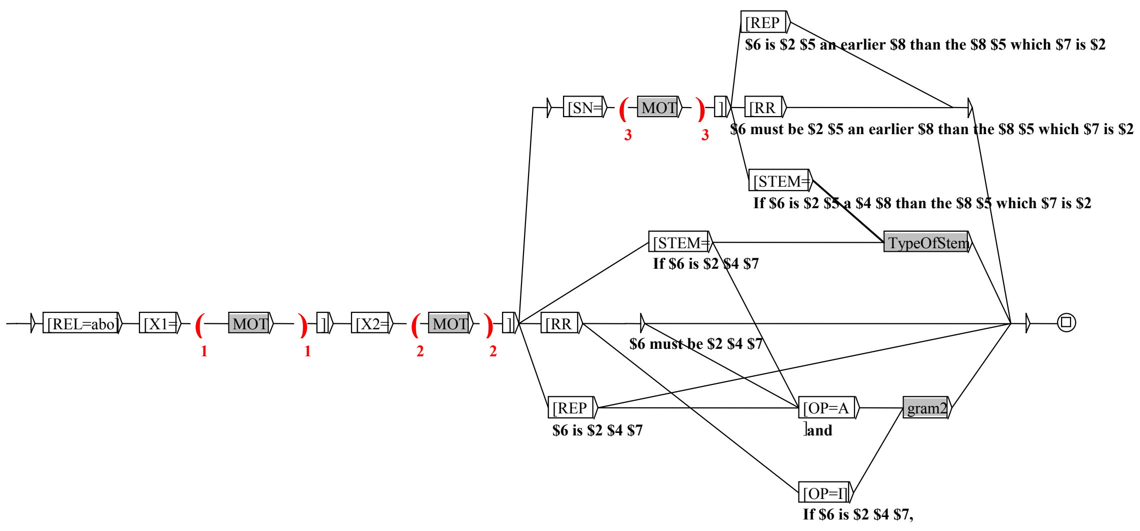

[REP
$6 is $2 $5 an earlier $8 than the $8 $5 which $7 is $2
[SN=
MOT
3
3
[RR
$6 must be $2 $5 an earlier $8 than the $8 $5 which $7 is $2
[STEM=
If $6 is $2 $5 a $4 $8 than the $8 $5 which $7 is $2
[STEM=
If $6 is $2 $4 $7
TypeOfStem
[REL=abo
[X1=
MOT
1
1
[X2=
MOT
2
2
[RR
$6 must be $2 $4 $7
[REP
$6 is $2 $4 $7
[OP=A
and
gram2
[OP=I
If $6 is $2 $4 $7,

Figure 6. Simplified view of one 'finite state template'

The role of the transducers is to a) identify patterns in the intermediary text representation (displayed in Figure 5) ; b) to fill some variables with parts of the matched text (delimited by the large parenthesis in the graph) and c) to produce an output. The output of the graph is the text written below the boxes (see Fig. 6). For example, the Fig. 6 graph applied on the Fig. 5 intermediary text will match the first line of the text (which is an ABO rule) and produce the output "$6 must be $2 $4 $7" where *$x* variables are replaced by their content (the corresponding "words" identified in the intermediary text representation).

5 Conclusion

Template-based methods are often perceived as a poor procedure for full-scale NLG systems. For example, Langklide (1998) says that "templates only work in very controlled or limited situations" and that they cannot provide "the expressiveness, flexibility or scalability that many real domains need". Despite this perception our investigations demonstrated that they offer accurate control on the generated text (a crucial element in the context of test development), they are easy to build and to adapt to different languages. So, as Reiter (1995) has previously concluded, we decided that "NLG shouldn't get in the way" and we have opted for a goal-oriented approach. This approach decidedly fulfills the "real domain needs" we have defined and does so in a less costly manner that is easier to monitor at every step of the processing. In summary, the ATG system is proving to be quite adequate for the purpose of generating text for AR items and is a vital component of a fully functioning automatic item generation system. Such a system would be of significant utility to large scale testing programs by:

- Dramatically increasing item production capability to permit replacement of retired items;
- Refining the quality of items for administration by implementing generation controls on the language used for the item;
- Provision of a priori statistical characteristics of items, thus permitting completely autonomous test generation on-the-fly, even for Computerized Adaptive Testing.

Together, these automated item generation elements (abstract representation, natural language generation, and difficulty modeling) offer tantalizing possibilities for the future of test production, made possible by the contributions of this ATG system.[7]

References

Brown James Dean (1997), "Computers in language testing: present research and some future directions". *Language Learning & Technology*. Vol.1. N°1: 44-59.

Dalianis, H. (1999), "Aggregation in Natural Language Generation". *Journal of Computational Intelligence*, Vol. 15, N°4: 384-414.

Dennis, I., Handley, S., Bradon, P., Evans, J., and Newstead, S.E. (2002), "Approaches to modeling item generative tests". In Kyllonen, P. and Irvine, S.H. (Eds.) *Item Generative Testing*.

Fairon, Cédrick, ed. (1998-1999), *Analyse lexicale et syntaxique: Le système INTEX*, Lingvisticae Investigationes Tome XXII (Volume spécial), Amsterdam/Philadelphia: John Benjamins.

Gross, Maurice (1997), "The Construction of Local Grammars", in E.Roche et Y.Schabes (eds.), *Finite State Language Processing, Cambridge, Mass*., The MIT Press: 329-352.

Langklide Irene and Knight Kevin (1998), "Generation that Exploits Corpus-Based Statistical Knowledge". In *Proc. COLING-ACL*.

Reiter, Ehud and Robert Dale (1997), "Building Applied Natural Language Generation Systems". *Natural Language Engineering*, 3: 57-87.

Reiter, Ehud (1995), "NLG vs. Templates". In *EWNLG*, Leiden.

Silberztein, Max (1993), *Dictionnaires électroniques et analyse automatique de textes.* Le système INTEX, Paris, Masson.

Silberztein, Max (2000), "INTEX at IBM". In Anne Dister (ed). *Proceedings of the Third INTEX workshop*, RISSH. Université de Liège.

Wainier, H. (1990), *Computerized Adaptive Testing: A Primer*. Lawrence Erlbaum Associates: Hillsdale, New Jersey.

Dunkel Patricia. (1997), "Computer-Adaptive Testing of Listening Comprehension: A Blueprint for CAT Development". In *The Language Teacher Online*. JALT.

7 The ATG system is currently being integrated into a "Test Creation Assistant" (TCA) to be used by test developers at ETS. Examples of automatically generated items are published on-line at:
http://www.fltr.ucl.ac.be/FLTR/TEDM/ets.html