A computer-aided environment for construction of multiple-choice tests

Ruslan Mitkov,
Le An Ha
Jon Bernardes
School of Humanities, Languages, and Social Sciences

Background and rationale

Multiple-choice tests have proved to be an efficient tool for measuring students’ achievement and are used on a daily basis both for assessment and diagnostics worldwide. Statistics suggest that the Question Mark Computing Ltd.’s testing software Perception alone has had more than 3,000,000 users so far, of which at least 95% have taken multiple-choice question tests.1 E-learning has created an even higher demand for multiple-choice questions as they are one of the most suitable ways for an e-learner to get feedback.

Each multiple-choice test item consists of a stem – a question or incomplete statement (e.g. ‘Who was voted the best football player for 2004?’, correct answer (in this case ‘Ronaldinho’) and distractors (incorrect, but desirably plausible choices such as ‘Henry’, ‘Ronaldo’ and ‘Rooney’ in this example). The manual construction of such tests, however, is a time-consuming and labour-intensive task. One of the main challenges in constructing a multiple-choice test is, among other things, the selection of plausible alternatives to the correct answer which will better distinguish confident students from unconfident ones.

The objective of this project was to provide an alternative to the lengthy and demanding activity of developing multiple-choice tests and propose a new, Natural Language Processing (NLP) based approach for generating tests from instructional texts (textbooks, encyclopaedias). Using various NLP techniques, the system identifies key concepts in the text (on which test items will be based) and sentences containing essential information. It then converts a sentence (statement) into a question, using transformation rules, and produces distractors, which have close meanings to the correct answer (identified using different NLP-based similarity measures). Work on a pilot project has shown that the proposed semi-automatic procedure is up to 3.8 times quicker than a completely manual one (Mitkov and Ha, 2003).

The innovation / research

The main innovation of this research project is its original methodology which employs various NLP techniques to automatically generate multiple-choice test items from electronic textbooks/documents. This methodology is based on the premise that test items should focus on key concepts rather than addressing less central and even irrelevant concepts or ideas. Therefore, the first stage of the procedure is to identify domain-specific terms which can serve as ‘anchors’ of a test item. By way of example, the word ‘syntax’ is a prime candidate for a domain-specific term in the sentence:

“Syntax is the branch of linguistics which studies the way words are put together into sentences”.

This sentence can then be transformed into a question about this term such as:

“Which branch of linguistics studies the way words are put together into sentences?” or

“Which discipline studies the way words are put together into sentences?”

both of which can act as stems in multiple-choice test items.

1 Personal communication from Question Mark Computing Ltd.
Another premise is that distractors should be as semantically close to the correct answer as possible so that no additional clues are provided for the students. Semantically close distractors are more plausible and therefore better at distinguishing good, confident students from poor and uncertain ones. In the above example, the distractors for the correct answer *syntax* should preferably be *semantics* or *pragmatics* instead of *chemistry* or *football*.

In order to keep the test item comprehensible and avoid additional complexity, a stem is generated from a declarative finite clause, as in the example above, using simple transformational rules which, in turn, results in only minimal change of the original wording. We will refer to the clause that gives rise to the stem of a test item as the *source clause*.

Underpinned by the above principles, a system for computer-aided generation of multiple-choice test items (each featuring three distractors) from instructional documents in electronic form has been implemented. The system is built on separate components, which perform the following tasks: (i) term extraction, (ii) stem generation and (iii) distractor selection (Diagram 1).

![Diagram 1: Structure of the system](image)

### Outcomes and benefits

The main outcome of this project is the implementation of a multiple-choice question generation which takes electronic texts as input, and produces multiple choice test items. The items generated with the help of the system are declared by the post-editor (usually the module leader / examiner) as either a) ‘unworthy’ and to be discarded or b) ‘worthy’ and either to be accepted without any revision or to be post-edited (e.g by changing the wording, replacing distractors, etc.) before being put into use. The post-editor is instructed to mark items which require too much revision or do not ask about a central concept as unworthy. The items selected for further post-editing require minor, fair or major revisions. To assist the post-editor in their task, a user-friendly environment was developed for the second post-editing exercise. This environment is web-based and communicates with an SQL (structured query language) database where the revisions undertaken by the post-editors are recorded. For example, if the test item originally produced by the system is:

(4) What do words and phrases form?

   i. the constituents of the clause
   ii. the phrases of the clause
   iii. the sentences of the clause
   iv. the optional constituents of the clause
The user-friendly environment would provide the second post-editing options shown in Figure 1 below.

![Figure 1. A snap-shot of the post editing environment](image)

The system generates a list of distractors for each test item that have to be approved by the post-editors. The users select a distractor in a post-editing environment by ticking the box next to it. Inappropriate distractors are deselected by unticking the box next to them. As illustrated in Figure 1, the post-editor has selected the distractors “central constituents” and “elements” from the proposed list to replace “phrases” and “sentences” originally generated. If none of the alternatives satisfies her, she can type the distractor of her choice in the “manually input” field.

The post-editing environment includes additional functionalities such as listing the sentence which contains the source clause that gave rise to the stem of the test item, in this case “Just as words and phrases form the constituents of the clause rank, so […]”. It also provides fields for revising the stem as well as the left and the right context of the answer and the distractors. These fields consist of words which appear in the source clause and were originally selected to accompany the answer (rather than the stem) by the system. For instance, the right context of the answer in (4) originally contained the phrase “of the clause” but not the word “rank”. As Figure 1 shows, the post-editor added this word to the right context field. This word now appears in the answer as well as the distractors that accompany the stem of the test item.

After the post-editor has finished revising a given item, she has to click on the submit button. Then, the system stores the post-edited item along with information about the level of editing in the database. Each test item can be accessed by more than one post-editor at the same time and the revisions are stored separately for each editor.

The generic multiple-choice generation system was applied to the areas of linguistics and sociology which were identified as the teaching foci of the current CELT project. The aforementioned post-editing environment was developed to serve lecturers who were invited to submit their textbook, see what the system can generate and post-edit, using the web interface. ([http://celg.wlv.ac.uk/Web-based-MCQ/submitTextBook.php](http://celg.wlv.ac.uk/Web-based-MCQ/submitTextBook.php)).

Given that the manual construction of multiple-choice items is a time-consuming and labour-intensive task, the proposed work will have a significant impact both in terms of methodology and practice. It could be of benefit to both lecturers when developing multiple-choice tests for assessment purposes as well as for students who would like to use these tests for feedback. For the first time a new, original semi-automatic methodology will assist the production of multiple-choice test items. This, in turn, will have a dramatic effect on cutting the cost of developing multiple-choice tests and, as a result, will provide new opportunities in extending the pools of available tests, thus diversifying (and improving the quality of) the everyday assessment task.
Evaluation

The methodology was evaluated in two ways. Firstly, we investigated the efficiency of the procedure by measuring the average time needed to produce a test item with the help of the program as opposed to the average time needed to produce a test item manually. Secondly, we examined the quality of the items generated with the help of the program, and compared it with the quality of the items produced manually. The quality was assessed via standard test theory measures such as discriminating power and difficulty of each test item, and the usefulness of each alternative was applied. In the next two sections we report on our evaluation efforts using these measures.

(i) Efficiency of computer-aided test item generation

For each post-editing exercise, we calculated the overall time it took for the post-editors to perform the post-editing task, including rejecting unworthy items. In the first post-editing exercise, this time was 540 minutes which, divided by the number of 328 worthy items (as these items represent the end-product of the whole procedure), yielded an average of 1 minute and 36 seconds per worthy test item to be constructed.

After the first post-editing exercise the two post-editors involved were asked to produce manually 65 test items from another chapter of the Linguistics textbook. A different chapter than the one used as the input to the system was chosen to ensure that the post-editors were not familiar with its content or biased by the post-editing task they had recently undertaken. However, being part of the same textbook, the writing style and the difficulty of this chapter was similar to the one used as the input to the system. This task took the post-editors 450 minutes resulting in an average 6 minutes and 55 seconds per manually constructed test item. Clearly, the average time the post-editors spent on producing items using the computer-aided procedure during the first post-editing exercise compares favourably with the average time they spent when producing test items manually.

In the second post-editing exercise the third post-editor produced 90 worthy items within 150 minutes with the help of the user-friendly environment. This averages to 1 minute and 40 seconds per worthy item, which is very similar to the time recorded for the post-editors of the first post-editing exercise.

The same person was asked to manually produce 40 test items from the chapter that was used as the input to the system. This was done more than a year after she performed the post-editing task so any effect of familiarity with the material was practically extinguished. It took the third post-editor 240 minutes to manually produce these test items which results in an average of 6 minutes per item. This average is comparable to the time recorded for her colleagues in the first post-editing exercise.

Therefore, although using the post-editing environment did not result in faster post-editing times for the third post-editor when compared to her two colleagues from the first post-editing exercise, her average time per worthy item continues to compare favourably with the time it took her to produce items manually. Hence, computer-assisted test item generation is shown to be much faster than manual production of test items in both post-editing exercises.

<table>
<thead>
<tr>
<th>Method Description</th>
<th>Produced items</th>
<th>time mins</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer-aided without post-editing environment</td>
<td>328</td>
<td>540</td>
<td>1 min 36secs</td>
</tr>
<tr>
<td>Manual – different chapter</td>
<td>65</td>
<td>450</td>
<td>6 min 55secs</td>
</tr>
<tr>
<td>Computer-aided with post-editing environment</td>
<td>90</td>
<td>150</td>
<td>1 min 40secs</td>
</tr>
<tr>
<td>Manual – same chapter</td>
<td>40</td>
<td>240</td>
<td>6 mins</td>
</tr>
</tbody>
</table>

Table 1. Efficiency of the post-editing method
(ii) In-class experiments

Controlled sets of the post-edited test items were used to test students and obtain evaluation data related to the items' quality. Only items approved by a linguistics lecturer were used in the experiment (i.e., it was made sure that the items addressed material taught to the students). Two experiments were conducted. In the first experiment, 24 test items constructed in the first post-editing exercise were employed (produced without the post-editing environment; referred to in Table 2 below as ‘first computer-aided’). Another 12 manually produced items were also included in this experiment.

In the second experiment, 18 items produced in the second post-editing exercise were employed (constructed with the help of the post-editing environment; referred to in Table 2 below as ‘second computer-aided’). The same 12 manually produced items which were used in the first experiment, were included in the second experiment as well.

The generated tests were delivered via Questionmark’s Perception testing software which, in addition to providing a user-friendly interface, computes diverse statistics related to the test items answered. Perception has a web-based interface which makes the test accessible to the students no matter where they are located. The test item displayed (Figure 2) is one of the post-edited items that were ported to the Perception software. The position of the correct answer is randomly generated.

![Figure 2. Test-item delivered](image)

The first experiment was conducted in class and the participants were supervised. In the second experiment, the students accessed the interface from their own web browser and conducted the test without supervision. In both cases, the software would accept answers only from the students who completed the task within 30 minutes. We acquired data from 30 undergraduate students in linguistics for the first experiment and 78 students for the second.

The current experimental setting does not look at the problem of delivering a balanced test of preset overall difficulty based on random (or constraint-driven) selection of test items. Instead, it focuses on exploring the feasibility of the computer-aided procedure and on the quality of the test items produced.

(iii) Analysis of post-edited test items

Item analysis is an important procedure in classical test theory which provides information as to how well each item has functioned. The item analysis for multiple-choice tests usually consists of the following information (Gronlund, 1982): (i) the difficulty of the item, (ii) the discriminating power and (iii) the usefulness of each distractor. This information can tell us if a specific test item was too easy or too hard, how well it discriminated between high and low scorers on the test and whether all of the alternatives functioned as intended. Such types of analysis help improve test items or discard defective items.

---

2 Originally called ‘effectiveness’. We chose to term this type of analysis ‘usefulness’ to distinguish it from the (cost/time) ‘effectiveness’ of the semi-automatic procedure as opposed to the manual construction of tests.
In order to conduct this type of analysis, we used a simplified procedure, described by Gronlund (1982). We arranged the test papers in order from the highest score to the lowest score. We selected one third of the papers and called this the upper group (10 papers in the first experiment, 26 in the second). We also selected the same number of papers with the lowest scores and called this the lower group. For each item, we counted the number of students in the upper group who selected each alternative; we made the same count for the lower group.

**Item Difficulty**

We estimated the *Item Difficulty* (ID) by establishing the ratio of students from the two groups who answered the item correctly (ID = C/T, where C is the number who answered the item correctly and T is the total number of students who attempted the item). As Table 2 shows, from the 24 items in the first experiment (produced in the first post-editing exercise) subjected to analysis, there were 0 too difficult and 3 (12.5%) too easy items. The average item difficulty was 0.75. From the 18 items produced in the second post-editing exercise with the help of the web-based environment and tested in the second experiment, none were too easy or too difficult with the average item difficulty being 0.58, which is closer to the recommended score of 0.50 for ID.

**Discriminating Power**

We estimated the item’s *Discriminating Power* (DP) by comparing the number students in the upper and lower groups who answered the item correctly. It is desirable that the discrimination is positive which means that the item differentiates between students in the same way that the total test score does. The formula for computing the *Discriminating Power* is as follows: DP = (Cu – Cl)/T/2, where Cu is the number of students in the upper group who answered the item correctly and Cl the number of the students in the lower group that did so. Here again T is the total number of students included in the item analysis. The average DP for the first set of items used in the class test was 0.40 and the average DP for the second set was 0.36. From the analysed test items, there was only one item that had a negative discrimination in the first set and none in the second. Hence, the items in both experiments did not differ a lot with respect to their DP.

<table>
<thead>
<tr>
<th>test</th>
<th>item difficulty</th>
<th>item discriminating power</th>
<th>usefulness of distractors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#items</td>
<td>#students</td>
<td>avg. item difficulty</td>
</tr>
<tr>
<td>First</td>
<td>24</td>
<td>30</td>
<td>0.75</td>
</tr>
<tr>
<td>computer-aided</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Second</td>
<td>18</td>
<td>78</td>
<td>0.58</td>
</tr>
<tr>
<td>computer-aided</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual</td>
<td>12</td>
<td>108</td>
<td>0.56</td>
</tr>
</tbody>
</table>

**Table 2. Item analysis**

The *usefulness of the distractors* is estimated by comparing the number of students in the upper and lower groups who selected each incorrect alternative. A good distractor should attract more students from the lower group than the upper group. The evaluation of the distractors estimated the average difference between students in the lower and upper groups to be 1.92 in the first set of questions and 2.94 in the second. Distractors classed as poor are those that attract more students from the upper group than from the lower group, and there were 6 such distractors produced in the first post-editing exercise (9.2% of all distractors produced in this exercise) and 3 (5.5%) in the second.

---

1 For experimental purposes, we consider an item to be ‘too difficult’ if ID ≤ 0.15 and an item ‘too easy’ if ID ≥ 0.85.
2 Zero DP is obtained when an equal number of students in each group respond to the item correctly. On the other hand, negative DP is obtained when more students in the lower group than the upper group answer correctly. Items with zero or negative DP should be either discarded or improved.
3 Maximum positive DP is obtained only when all students in the upper group answer correctly and no one in the lower group does. An item that has a maximum DP (1.0) would have an ID 0.5; therefore, test authors are advised to construct items at the 0.5 level of difficulty.
4 7 test items had only 2 distractors assigned.
On the other hand, we term distractors not useful if they are selected by no student. The evaluation showed that there were 3 distractors (4.6%) deemed not useful in the first set of items and 1 (1.8%) in the second. Hence, the items used in the second experiment were found to score considerably better with respect to their usefulness of the distractors (in addition to their average item difficulty), compared to the items produced without the aid of the user-friendly interface.

(iv) Analysis of the items constructed manually

An experiment worthwhile pursing was to conduct item analysis of the manually produced test items and compare the results obtained regarding the items produced with the help of the program. The set of 12 manually produced items were subjected to the above three types of item analysis using data produced by the students in both experiments.

There were 0 too difficult and 2 (16.7% of all manually constructed test items) too easy items. The average item difficulty of the items was 0.56, which is very close to the recommended score and the score achieved by the items in the second experiment. The average discriminating power was assessed to be 0.26 and there were 0 items with negative discrimination. The evaluation of the usefulness of the distractors resulted in an average difference between students in the upper and lower groups of 1.18. There were 5 distractors (15.2%) that attracted more students from the upper group and were therefore, declared as poor and 8 (24.2%) distractors not selected at all, and therefore deemed to be not useful. Therefore, the items produced via the computer-aided procedure, especially when the user-friendly interface had been employed, were found to score better than the manually produced items as far as discriminating power and usefulness are concerned.

Discussion and plans for future work

The evaluation results clearly show that the construction of multiple-choice test items with the help of the program is much more effective than purely manual construction. We believe that this is the main advantage of the proposed methodology. As an illustration, the development of a test databank of considerable size consisting of 1000 items would require 30 hours of human input when using the program, and 115 hours if done manually. This has direct financial implications as the time and cost in developing test items would be dramatically cut.

At the same time, the test item analysis shows that the quality of test items produced with the help of the program is not compromised in exchange for time and labour savings. The test items produced with the program were evaluated as being of very satisfactory quality. As a matter of fact, in many cases they scored even better than those manually produced. However, whereas the item difficulty factor assessed for manual items emerges as better, of those produced with the help of the program, there were only 3 too easy items and 0 too difficult ones. In addition, whilst the values obtained for the discriminating power are not as high as we would have desired, the items produced with the help of the program scored much better on that measure and, what is also very important, is that there was only one item among them with negative discrimination (as opposed to 2 from those manually constructed). Finally, the analysis of the distractors confirms that it is not possible to class the manually produced test items as better quality than the ones produced with the help of the program. The test items generated with the help of the program scored better on the number of distractors deemed as not useful, were assessed to contain fewer poor distractors and had a higher average difference between students in the lower and upper groups.

Most of the work on this project has been completed, apart from implementing a refinement of one of the modules and running final experiments in October and November 2005. In addition to extending the set of test items to be evaluated and the samples of students taking the test, further work includes experimenting with more sophisticated term extraction techniques and with other more elaborate models for measuring semantic similarity of concepts. We would like to test the feasibility of using collocations from an appropriate domain corpus with a view to extending the

---

7 3 test items had only 2 distractors assigned.
8 Ideally, item difficulty should be around the mark of 0.5
choice of plausible distractors. We also envisage the development of a more comprehensive grammar for generating questions, which in turn will involve studying and experimenting with existing question generation theories. As our main objective has been to investigate the feasibility of the methodology, we have so far refrained from more advanced NLP processing of the original documents such as performing anaphora resolution and temporal or spatial reasoning which will certainly allow for more questions to be generated. Future work also envisages evaluation as to what extent the questions cover the course material. Finally, even though the agreement between post-editors appears to be a complex issue, we would like to investigate it in more depth. This agreement should be measured on semantic rather than syntactic principles, as the post-editors may produce syntactically different test questions which are semantically equivalent. Similarly, different distractors may be equally good if they are equal in terms of semantic distance to the correct answer.

A suggestion that arose while experimenting in the domain of sociology, was that questions about several sentences should be generated. This is a challenging task, and could not be done within the scope of this CELT project. This would be a future ambitious task involving co-reference resolution (not only at NP level, but also at event level).

Future dissemination and embedding plans include the development of a web-service, and necessary components enabling users to get multiple-choice test items delivered automatically via their usual editors (i.e. Microsoft Word) rather than having to go to the website. This will further facilitate the use of the tool.

Conclusion

This CELT project report describes a novel NLP-based and computer-aided procedure for the construction of multiple-choice tests from instructional documents in electronic form. The results from the evaluation conducted suggest that the new procedure is very effective in terms of time and labour, and that the test items produced with the help of the program are not of inferior quality to those produced manually.

References


