

LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 10 : 9 September 2010

ISSN 1930-2940

Managing Editor: M. S. Thirumalai, Ph.D.

Editors: B. Mallikarjun, Ph.D.

Sam Mohanlal, Ph.D.

B. A. Sharada, Ph.D.

A. R. Fatihi, Ph.D.

Lakhan Gusain, Ph.D.

K. Karunakaran, Ph.D.

Jennifer Marie Bayer, Ph.D.

S. M. Ravichandran, Ph.D.

Technology Note Creating Parallel Test Items with Microsoft Excel

Renu Gupta, Ph.D.

Abstract

This note describes how to generate several similar test items (called parallel items) using Microsoft Excel. The simple code can be applied to several items, thereby simplifying the task of creating multiple versions of a test.

Introduction

In certain test situations, multiple versions of a test are required. For instance, one may need a back-up test for candidates who cannot take the test on the assigned date. In order to be fair to the test-takers but prevent cheating, the two tests contain different items but need to test the same construct. Considerable sophisticated work has been done on ensuring that parallel test items are equivalent (Boekkooei-Timminga, 1990; Brummel et al., 2008; Chen et al. 2008), but this note describes an extremely limited case where the test items are known to be parallel but we need to mechanically generate hundreds of parallel items.

The scenario described here is a typical English language proficiency test that examines knowledge of English grammar. Although there are significant problems with such a test (and its relationship to language proficiency), these tests are commonly used in India as a

proxy for language proficiency. The test contains discrete items such as tenses, prepositions, and subject-verb agreement, and a typical test item is as under:

Select the correct answer:
Sachin in Delhi now.
(a) live (b) lives (c) lived

It is possible to create a parallel version of this item by altering non-essential words around the target. Possible versions are:

- Rashid in Hubli now
- Mita in Kolkata now.

These are called parallel test items and each student gets only **one** item from the above. In such items, the basic construct remains the same; we are testing knowledge of tenses + subject-verb agreement. However, the padding around the item is trivial; it does not matter if the person is Sachin or Rashid or if the place is Delhi or Hubli. Parallel test items allow the tester to use different items with different students to test the same construct.

If we alter too many words, there is a danger of creating a non-parallel item that would require sophisticated comparison.

- Rashid in Hubli.

In this item, one would have to change the options, since two options become correct.

Background

For one assignment I was given a grammar test of 75 items and asked to create 750 similar items within a few days. The items were of the type described above.

The test was to be administered online, but since the students numbered in the hundreds, the issue of cheating arose. To solve this problem, the IT department has a solution: it randomized the order of the items. So, each student sees the 75 questions in a different sequence, and cannot memorize a fixed sequence of answers. To further ensure that students did not merely memorize the answers, I was asked to generate 10 versions of each test item, i.e., to create parallel items.

Generating Test Items

Typing 10 versions of ‘Sachin ... in Delhi now’ and then doing the same for the 75 items would be tedious. I asked the IT department to write a piece of code to do this

Language in India www.languageinindia.com

10 : 9 September 2010

Renu Gupta, Ph.D.

Technology Note - Creating Parallel Test Items with Microsoft Excel

automatically, but they required two days to come up with the code. So, I turned to Microsoft Excel, which has a formula bar. Using two functions—*concatenate* and *substitute*—I generated 10 versions of a test item. I then copied the same code to the other 74 test items and generated 740 test items in a few days.

Concatenate merely joins the parts of a sentence. *Substitute* works like *Find and Replace* in Microsoft Word. By using the two formulae, one can create all possible combinations of a sentence string.

Using the example above, we begin with the following:

Sachin in Pune now.
 Rashid in Hubli now.
 Mita in Delhi now.

This eventually yields the following items:

Sachin..... in Pune now.
 Sachin..... in Hubli now.
 Sachin..... in Delhi now.
 Rashid..... in Pune now.
 Rashid..... in Hubli now.
 Rashid..... in Delhi now.
 Mita..... in Pune now.
 Mita..... in Hubli now.
 Mita..... in Delhi now.

Procedure

Step 1

Divide the test item into three components—beginning, target, and end—and type these in three columns in Microsoft Excel.

	A	B	C
1	Name	Base	Place
2	Sachin	in Pune now.

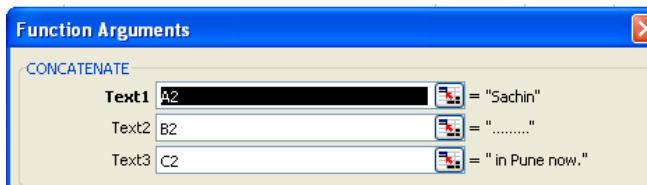
Step 2

In Columns 1 and 3, type in your alternatives.

	A	B	C
1	Name	Base	Place
2	Sachin	in Pune now.
3	Rashid		in Hubli now.
4	Mita		in Delhi now.

Step 3

Use a new column, Column D, to create the test items. In the formula bar, select the *Concatenate* formula and create a sentence from the pieces.

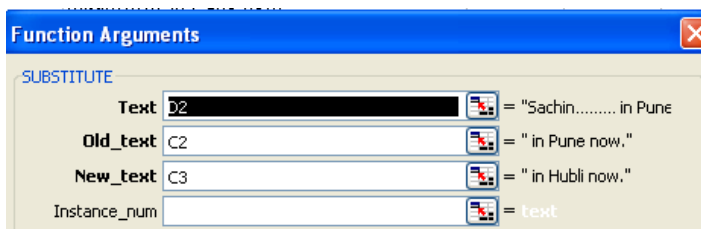


This gives the sentence below.

	A	B	C	D
1	Name	Base	Place	
2	Sachin	in Pune now.	Sachin..... in Pune now.

Step 4

Create a different combination using the *Substitute* function.



This will give you the sentence below.

	A	B	C	D
1	Name	Base	Place	
2	Sachin	in Pune now.	Sachin..... in Pune now.
3	Rashid		in Hubli now.	Sachin..... in Hubli now.

Step 5

Select the latest sentence and drag the cursor down. The formula gets copied to the next row, giving you a new combination.

D
Sachin..... in Pune now.
Sachin..... in Hubli now.
Sachin..... in Delhi now.

Step 6

Repeat Steps 3 to 6 for each item in Column 1. The result is the following set of sentences.

Sachin..... in Pune now.
 Sachin..... in Hubli now.
 Sachin..... in Delhi now.
 Rashid..... in Pune now.
 Rashid..... in Hubli now.
 Rashid..... in Delhi now.
 Mita..... in Pune now.
 Mita..... in Hubli now.
 Mita..... in Delhi now.

Step 7

You now have a piece of code in Column D. You can view this code by clicking on each sentence; the code shows up in the formula bar.

You can use this piece of code to create similar items. In a separate worksheet, create a new test item as in Steps 1 and 2. Copy the code you have created and use *Paste Special* to paste it into the new worksheet. New test items are automatically generated.

			Sentences
Some members of the cricket team	leaving	Some members of the cricket team..... leaving
Some members of the class	absent	Some members of the cricket team..... absent
Some members of the hockey team	not here	Some members of the cricket team..... not here
		Some members of the class..... leaving
			Some members of the class..... absent
			Some members of the class..... not here
			Some members of the hockey team..... leaving
			Some members of the hockey team..... absent
			Some members of the hockey team..... not here

The screens above show the most basic manipulations of strings to create nine parallel sentences. Additional sentences can be generated by adding text in Columns 1 and 3. More complex combinations can be created by manipulating the options in Column 2, or creating additional columns.

Conclusion

This note described a simple method to generate multiple versions of a test item. It can be used only in limited cases that allow non-essential information to be manipulated. One case is grammar items in an English language test, but the method cannot be extended to vocabulary or reading comprehension.

References

- Boekkoei-Timminga, E. (1990). The construction of parallel tests from IRT-based item banks. *Journal of Educational Statistics*, 15, 2, 129-145.
- Brummel, B.J., Kupp, D.E., and Spain, S.M. (2009). Constructing parallel simulation exercises for assessment centers and other forms of behavioral assessment. *Personnel Psychology*, 62, 137-170.
- Sun, K-T., Chen, Y-J., Tsai, S-Y., and Cheng, C-F. (2008). Creating ITR-based parallel test forms using the Genetic Algorithm method. *Applied Measurement in Education*, 21, 2, 141-161.

Renu Gupta, Ph.D.
4 Cornwell Road
Richmond Town
Bangalore 560025
Karnataka, India
renu@stanfordalumni.org