FIRST - Bug #3856

Inconsistent functionality: Parse common choices and associate them with each of the pertinent questions

03/04/2009 12:46 PM - Sandeep Namilikonda

Status:	Resolved	**Start date:**	03/04/2009
Priority:	Normal	**Due date:**	
Assignee:	Sandeep Namilikonda	**% Done:**	0%
Category:	client	**Estimated time:**	0.00 hour
Target version:	post-clientprototype	**Spent time:**	0.00 hour
Bugzilla-Id:	3856		

Description

Exam questions sometimes are grouped under a sub-category based on a common set of responses applicable to each of them. In such cases, the exam would have text preceding that subset of questions, in which case, its the parser's job to scan through the text, associate those common choices to each of the questions in that set, and finally, remove that portion of text.

Right now, this functionality is flaky (e.g., works on doc1.pdf but fails on doc10.pdf). Find out the reason for the glitch and fix it!

History

#1 - 03/26/2009 10:45 PM - Sandeep Namilikonda

In PDFAssessment2_4.java, separateQuestions() and combineAndSplit() functions are executed one after the other, which are responsible for the glitch. Here is what I learnt from the code:

separateQuestions():

```
instruction block is identified based on the presence of "[i-j]" or
    "i through j" or "question i and j"

For each question included in the question-range, instruction block is
    prepended.
```

combineSplitQuestions():

```
If the text (instructions) preceding a group of questions is found to
        contain options then those options are associated with each of the
        questions in that group!

a) First, options are saved when such case is recognized.
    b) Then, for each subsequent question recognized without options,
       the stored options get associated! (appended)
    c) The common text deemed as instructions preceding the options
           is deleted!
```

The problem occurs in combineSplitQuestions() when a document has an
instruction block (e.g., "questions 1-5" or "questions 1 through 5")
followed by options for those questions also included in the instruction
block. In this case, separateQuestions() first associates the entire instruction block, including the options, with each of the questions 1 through 5. Then, combineSplitQuestions() looks at the text for question 1, saves the options
text and deletes the text, assuming that the text is only an instruction block that precedes a list of questions! Unfortunately, in this case, this operation results in deleting the question itself and hence, the final result seems to miss the questions 1-5!

Only two cases of text preceding a bunch of questions are currently distinguished and handled.

a) question numbers for which the common instructions are applicable are
specified in the text, e.g., "questions 1-8 ".

```
In this case, separateQuestions() would have marked the instructions
   as an 'INSTRUCTION BLOCK' and would associate with each of
   the questions in the range: 1-8!
```

b) question numbers aren't explicitly identified but common options are
specified, in which case,
the options are nicely associated with each question that
follows the common instructions until a question with its own
options is discovered. In this case, the common instructions

are rightly 'removed' (e.g., doc1.pdf)!

Hence, make amends in SeparateQuestions() first, to look for "options"
text within the common instructions block. If options found then
segregate options from common instructions and strip off the
question number qualifier so that combine-split will correctly
associate options to questions that follow.

#2 - 03/31/2009 12:39 PM - Ryan McFall

Here's what Sandeep proposes:

Hence, make amends in SeparateQuestions() first, to look for "options"
text within the common instructions block. If options found then
segregate options from common instructions and strip off the
question number qualifier so that combine-split will correctly
associate options to questions that follow.

I think that this makes some sense. However, I can still think of cases where it can be problematic. What if the common instructions contain MORE than just the options to be used? How will we know which parts of the common instructions should be kept, and which part should be stripped out? In the example you've cited in doc10.pdf, how do we know that the text "select the correct answer from the following choices" should not be kept, for example? What if there were other instructions that should be kept, but the choices should not?

Probably what we want to happen is to not delete valid questions. You solution probably points us in the right direction for this to happen, at the expense of possibly removing some common instructions. I guess I feel like that's a decent trade-off. If you want to give this a shot and see what happens, go ahead.

#3 - 05/01/2009 05:13 AM - Ryan McFall

Sandeep, this bug is marked as client-alpha3. Can you look at it and either resolve it if you think it's fixed, or give an opinion as to whether it should be client-prototype or postclient-prototype? Thanks.

#4 - 05/02/2009 07:36 AM - Sandeep Namilikonda

Ryan, I made necessary changes in code to implement the solution I suggested under "Comment #1". But, as you pointed out in "Comment#2", there may be cases on which the solution implemented would end up pruning content from the common instructions block. I do not know of a document or a test case that actually fails the current solution. So, I am going to mark this bug as
postclient-prototype and set its resolution to REMIND.

Is that agreeable to you?

#5 - 05/02/2009 02:12 PM - Ryan McFall

This resolution seems fine to me

#6 - 03/27/2013 02:24 PM - Redmine Admin

Original Bugzilla ID was 3856