IRR ODL
THE INTERNATIONAL
REVIEW OF RESEARCH IN
OPEN AND DISTANCE LEARNING

Identification of Conflicting Questions in the PARES System

Avgoustos Tsinakos and **Ioannis Kazanidis**
Kavala Institute of Technology, Greece

Abstract

Student testing and knowledge assessment is a significant aspect of the learning process. In a number of cases, it is expedient not to present the exact same test to all learners all the time (Pritchett, 1999). This may be desired so that cheating in the exam is made harder to carry out or so that the learners can take several practice tests on the same subject as part of the course.

This study presents an e-testing platform, namely PARES, which aims to provide assessment services to academic staff by facilitating the creation and management of question banks and powering the delivery of nondeterministically generated test suites. PARES uses a conflict detection algorithm based on the vector space model to compute the similarity between questions and exclude questions which are deemed to have an unacceptably large similarity from appearing in the same test suite. The conflict detection algorithm and a statistical evaluation of its accuracy are presented. Evaluation results show that PARES succeeds in detecting question types at about 90% and its efficiency can be further increased through continuing education and enrichment of the system's correlation vocabulary.

Keywords: Computer adaptive testing; CAT; conflict detection algorithm

Introduction

In recent years, e-learning has made significant progress in every way that can be measured. Multiple e-learning platforms exist, both open source (Moodle, ILIAS, ATutor) and commercial (BlackBoard), and these have matured considerably over the years, offering comprehensive and powerful tools with which to facilitate the learning process.

Even though e-learning platforms and tools have displayed significant progress as outlined above, there still remain several aspects of the learning process that are not quite adequately provided for. One such significant aspect is student testing and knowledge assessment, more concisely known as e-testing. Most of the times platform's teaching staff has to maintain and administer large question banks covering a multitude of subjects, an endeavor which requires advanced software features that are still being discussed, refined, and developed. An especially thorny problem arises when question banks are exported and merged together with other, previously existing testing material. It is quite possible that these suites may not be altogether effective, fair, and without conflicts.

Effectiveness is a quality that cannot be objectively measured a priori, but we can make several effectiveness-related objective observations based on practical situations. One such observation is that in a sizable number of cases, it is expedient to not present the exact same test suite to all learners all the time (Pritchett, 1999). This may be desired so that cheating in the exam is made harder to carry out or so that the learners can take several practice tests on the same subject as part of the course. In any case, this aim can be achieved by randomly selecting questions from a question bank according to some predefined algorithm (Kikusawa et al., 2006).

The same inability to provide a waterproof objective definition is also encountered when discussing fairness, but in this case also we can constrain ourselves to a single aspect of it. Specifically, we would like to assert that each of these randomly generated test suites provides in each case a more or less constant (within some acceptable bounds) balance among the subjects it covers.

The final of these testing system requirements is today the most difficult to achieve. A conflict in a test suite is defined as the simultaneous presence of two or more questions that are redundant in content and/or one of their number provides a part or the whole of the answer for another (Hage & Aimeur, 2006).

This paper introduces PARES, a platform that is being developed in our institution to provide learning assessment tools closely tailored to our teachers' and professors' needs. The latest improvement in PARES, which is the main subject of this paper, concerns the integration of information retrieval (IR) techniques to identify conflicting questions in the question banks and prevent their mutual inclusion in the same test instance. This functionality is a specialized case of the search problem and uses keywords for each question to compute the similarities between questions using the cosine function in the vector space model (Salton et al., 1975). For additional efficiency, *term frequency/inverse document frequency* (*tf-idf*) weighting is applied to keywords when constructing question vectors.

The present paper is organized as follows: a brief description of basic information retrieval methods; a presentation of some related work; a presentation of PARES; details about the specific methodology used in PARES to generate random test suites with no conflicting questions; a presentation of the evaluation procedure and results; and, finally, a conclusion offering avenues for future work..

Information Retrieval

The function of IR is to provide easy access to information of interest to humans, typically given incomplete or even misleading user input, which is commonly referred to as the search query. The medium of user input and the nature of the stored information differs among several branches of modern information retrieval, such as full-text search, image retrieval, shape recognition, cross-language queries, and retrieval of human speech. To refer to the multitude of different types of information articles, henceforth, we will be using the general term document.

While initially it seems that finding the required information is the only task performed by an IR system, in fact today's large information corpora present another, not significantly easier, challenge: how to ascertain which of the multitude of search results better corresponds to the input data. Commonly this problem is solved by developing a method that assigns a relevance score to each document, according to which the documents are subsequently ranked. Several models used to compute and assign these scores have been developed through the years (Jiang, 2009), such as set-theoretic, probabilistic, and algebraic models.

Set-theoretic models represent documents as sets of terms. The relevance of each document to the search query is then derived from sequences of set-theoretic operations on these sets. The Boolean model of information retrieval is a classic example of this type of model and, at the same time, the first and most widely adopted one.

Probabilistic models treat the process of document retrieval as a probabilistic inference. Similarities are computed as the probabilities of each document being relevant for a given query. Probabilistic retrieval was initially proposed by Maron and Kuhns (1960), and to date several such models have been developed.

Algebraic models represent both documents and search queries as vectors or matrices. The similarity of each document with the search query is typically a scalar value calculated through some algebraic operation performed on them. The most well-known of these models is the vector space model, wherein documents are represented as vectors of scalar values. Each dimension of the vector corresponds to a separate term, which may be words, phrases, or other items depending on the application. Typically the values in the vector are positive for terms that occur in the document and zero for terms that do not, although other arrangements are also possible as there are several methods to compute the actual vector values (terms or weights). To determine the similarity between a document and the search query and thus provide ranked search results, vector space model implementations evaluate the similarity between the corresponding vectors. A common method to perform this evaluation is by computing the angle between the two vectors and regarding it as evidence of divergence; the cosine of this angle can then be used to provide a value in the range (0, 1) which corresponds to the relevance between document and query.

Related Work

A subset of e-learning belonging to the evaluation and assessment phase is e-testing. With this term we describe the whole lifecycle of authoring, delivery, and subsequent result analysis of testing material through electronic means. Today a lot of e-learning platforms such as Moodle, Blackboard, and others offer integrated tools to facilitate e-testing using various testing paradigms. A notable fact is that these tools are commonly perceived to not be able to fully take advantage of the strong points of e-testing and instead are perceived as in need of significant future development. Although evaluation and assessment is a very important part of e-learning, each platform offers e-testing functionality with limited functionality with which to attack important problems such as those mentioned earlier.

Specifically, each e-learning platform commonly implements its own authoring and delivery tools, uses a different storage format for the finished material, and has different requirements regarding the actual implementation of a working deployment. This means that there is considerable difficulty in reusing this material in a platform other than the one it was originally developed on and requiring considerable effort to translate and/or recreate the material as required in each case. As a result, the cost of e-testing material increases significantly and suboptimal use is made of the specialist effort required to produce it. The development of accepted standards for e-learning in general, such as ADL SCORM, and e-testing in particular such as IMS QTI (IMS GLC, 2011), is today a major step towards improved interoperability support.

In addition to the above, authoring tools and question pools for e-tests have become an integral and mandatory part of e-learning platforms. Some sophisticated platforms are reported in the literature, such as Plateau Exams (Plateau, 2011) and PARES (Kaburlasos et all, 2004). Moreover, some other systems, like AHA! and CAT-MD attempt to provide computerized adaptive tests (CAT). More specifically, AHA! (Romero et al., 2005) incorporate authoring tools that allow tutors to store their questions and create adaptive tests. Similarly, Triantafillou et al. (2008) implement a prototype called CAT-MD which provides CAT on mobile devices. In addition, Mustafa and Zualkernan (2010) use an adaptive method for selecting appropriate questions from various pools based on learners' answers to prior questions.

However, in dealing with e-testing and CAT issues, a new challenge arises, ensuring that created exams questions are free of conflicts. In many cases, conflict in an exam may exist when two or more questions are redundant in content and/or if one particular question reveals the answer of another question within the same exam. Question selection that depends on the teacher's preference cannot guarantee a flawless exam free of conflicts. For this reason, research has been conducted and new systems and tools have been implemented that attempt to detect these dependencies. More specifically, Bilenko and Mooney (2003) propose a framework for improving duplicate detection, using trainable measures of textual similarity, and Cadmus (Hage & Aimeru, 2006) uses information retrieval techniques to detect conflicts within an exam. For this reason, a module called ICE (Identification of Conflicts in Exams) is appended to the system, which is based on the vector space model relying on tf-idf weighing and the cosine function to calculate the similarity between questions.

PARES

PARES is an e-testing system that offers a comprehensive feature set targeted to managing testing and assessment in an academic environment. It includes tools to manage teachers and students, create logical courses and assign users to participate in or facilitate them, develop suitable testing material for each course, and administer tests to students according to a variety of testing paradigms. The results are then stored and made available in a variety of forms for further perusal. The platform is divided in three distinct modules, each one of which corresponds to a user role.

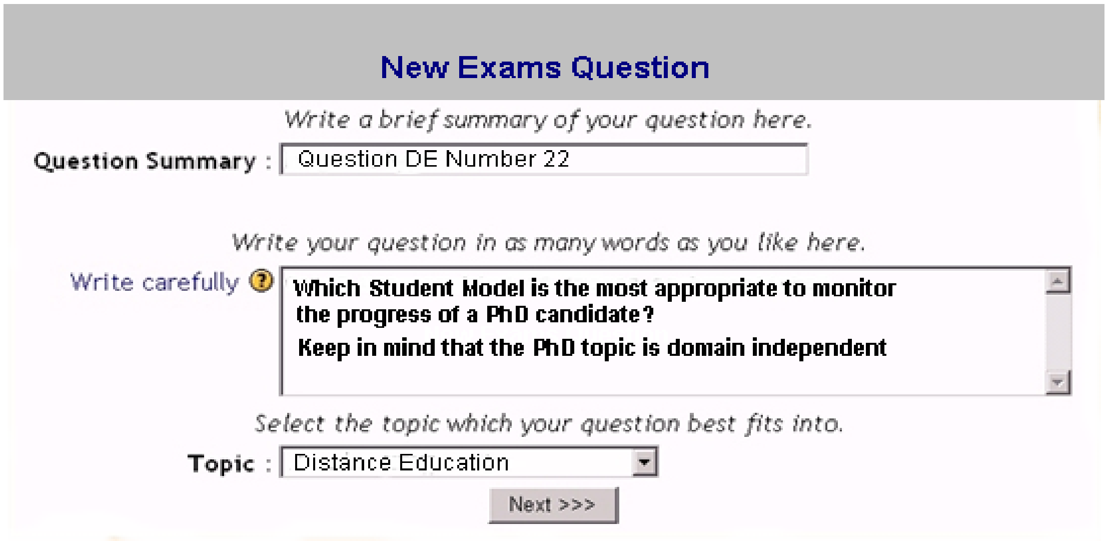
The administrator module in PARES allows the creation of user accounts and courses and the assignment of the former to the latter. The functions of this module are quite straightforward and commonly used in virtually all e-learning systems today; therefore, we shall not present it in greater detail here.

At the other end of the user spectrum is the PARES student module, which is used by learners to take tests electronically. Initially, these tests are constrained to multiple choice questions organized in question banks, from which tests are assembled. Learners may be allowed to take multiple tests, the significance of which is determined by the course teacher.

Finally, the most important module in PARES is used by teachers to develop testing material and determine the various test parameters.

Testing Material Development and Delivery

PARES offers teachers several tools to organize and develop testing material. Initially teachers submit new questions. The system prompts the teacher to provide a summary of the question and a description and to define the corresponded topic.



New Exams Question

Write a brief summary of your question here.

Question Summary : Question DE Number 22

Write your question in as many words as you like here.

Write carefully ?

Which Student Model is the most appropriate to monitor the progress of a PhD candidate?
Keep in mind that the PhD topic is domain independent

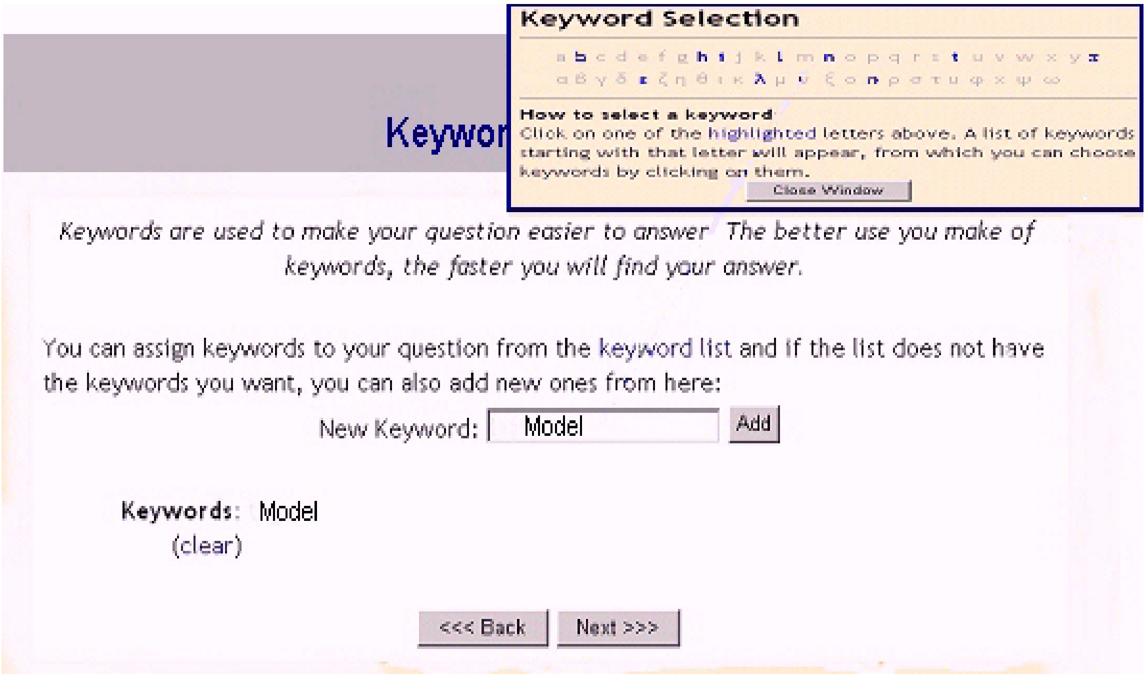
Select the topic which your question best fits into.

Topic : Distance Education

Next >>>

Figure 1. New exams question.

By clicking on the "Next" button, the keyword selection screen pops up (Figure.2). Selection of keywords can be done in two ways, either by manually typing them one at a time and selecting the "Add" option or by using the "keyword list" window, which displays all existing keywords.



Keywor

Keyword Selection

a b c d e f g h i j k l m n o p q r s t u v w x y z
α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω

How to select a keyword
Click on one of the highlighted letters above. A list of keywords starting with that letter will appear, from which you can choose keywords by clicking on them.

Close Window

Keywords are used to make your question easier to answer The better use you make of keywords, the faster you will find your answer.

You can assign keywords to your question from the keyword list and if the list does not have the keywords you want, you can also add new ones from here:

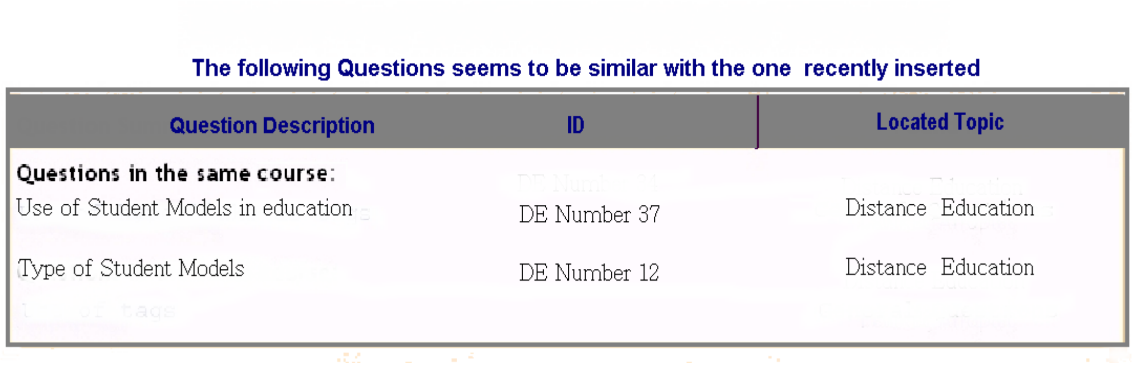
New Keyword: Model Add

Keywords: Model
(clear)

<<< Back Next >>>

Figure 2. Add keyword.

The user can also leave the keywords field blank and immediately click "Next." In this case, the system will automatically select which keywords describe this particular question using the built-in algorithm. Having the question submitted, the system tries to identify an existing question (similar case) that may provide an answer to the currently inserted question or which may be a replication of an already existing question. In case no similar questions are retrieved, the newly inserted question is stored in the database. On the contrary if one or more possible matches are found, a pop up screen displays the retrieved matches.



The following Questions seems to be similar with the one recently inserted

Question Description	ID	Located Topic
Questions in the same course:		
Use of Student Models in education	DE Number 37	Distance Education
Type of Student Models	DE Number 12	Distance Education

Figure 3. Potential similar questions.

By clicking on question summary, the user is allowed to view the full content of that specific question and conclude upon the potential replication. In such cases, the newly inserted question can be rephrased or even removed from the database. It is worth noting that in the case of a new defined keyword the domain expert is responsible to accept or reject it and to update the rejected keywords list. This is a list of words that will never be assumed valid keywords when the system automatically tries to assign keywords to a new question.

In order to better organize testing material each course in the system is assigned a curriculum by the teacher, which can be further broken down into chapters and units. The ability to associate several curricula with each course (only one of which may be active for a given teacher) allows different teachers to develop distinct approaches to testing the subject matter of a course. This is especially helpful when revising the testing methodology as it allows the system to continue functioning using the current methodology for a course while a newer one is being developed.

Each unit in a curriculum can be assigned a number of questions, the authoring of which is the responsibility of the teacher. Initially these questions are limited to multiple choice, but the underlying implementation allows different and more complex types of questions to be included in the future. Each question can furthermore be assigned a difficulty level. The teacher then can create test suites using the questions authored for each course (a question bank). To create a test, the teacher optionally selects a subset of the question bank within which the system will limit its activity and defines several important parameters such as a time limit, weights for testing each unit (i.e., how heavily it will be tested in relation to other units), penalties for wrong answers, and the desired difficulty for the test. When a student takes the test, the system automatically picks a suitable number of questions randomly, at the same time honoring the difficulty and unit weight limits set by the teacher. This increases the replay value of the test both among students, by making cheating harder, and also for each single student, by making each instance of the test unique.

Conflicts in Tests

An inherent problem with generating tests by randomly selecting questions within a question bank is that there will typically be several questions that are designed to assess the learner's knowledge on a single item. This state of affairs is practically guaranteed in PARES as it follows from the requirements of all test instances a) covering the same curriculum and b) being distinct. Therefore, for items belonging to a single unit we would like to avoid the possibility of including at the same time questions that are redundant or provide the answer to another included question in direct or indirect fashion. It can be argued that this is even a requirement from the system instead of a valuable feature. As a consequence, PARES contains logic specifically designed to avoid the inclusion of conflicting questions in the same exam. In order to detect such questions, the relevant subsystem uses IR techniques based on the vector space model. Within each set of interchangeable questions in the question bank, the similarities between questions are computed and constraints are placed on the maximum similarity between questions that can be chosen together.

Conflict Detection Algorithm

The conflict detection algorithm in PARES operates in two distinct phases: question authoring time and test generation time. During question authoring, each question is characterized according to teacher input by a set of keywords and/or keyphrases which must be present in the question body. The number of occurrences of each term in each question is calculated whenever a question is created or updated and stored in the system.

At test generation time, for each course unit the system retrieves the questions belonging to the union of two sets, these being a) the questions relevant to the unit and b) the questions that the teacher designated as usable in the current test. Since it is highly probable that there will be conflicting questions within this set, a document vector is computed for each question and the similarities between each pair are calculated according to these vectors. Question pairs with similarity above a certain threshold are deemed mutually exclusive and are treated by the system as such. Therefore, assuming satisfactory performance and a minimum number of questions in the bank, the resulting test is both randomized and free of conflicting questions.

Weight Calculation

The document vector for each question is multidimensional and contains the weights for each keyword or phrase (the word *term* will be used for these two kinds of text) that appears in at least one question in each course. The notation $w_{n,d}$ represents the weight for keyword *n* in the vector of document *d*, therefore:

$$V_d = \begin{bmatrix} w_{1,d} & w_{2,d} & \dots & w_{N,d} \end{bmatrix}^T$$

These weights are calculated according to the tf-idf weighting method. This method relies on term frequency and inverted document frequency to calculate each weight in turn as the product of these two factors. Term frequency (tf) represents the importance of term *n* in document *d* and is calculated as the quotient of the term *n's* frequency in document *d* divided by the maximum frequency among all terms appearing in said document. This is given in the next equation.

$$\boldsymbol{f}_{n,d} = \frac{f_{n,d}}{\max(f_d)}$$

The inverted document frequency (idf) serves as a metric of the discriminatory power of each term and is determined by an operation among all questions in a course. Higher values of *idf* therefore correspond to terms which characterize a question more distinctly than others. The *idf* is computed as the logarithm of the total number of questions divided by the number of questions in which term *n* appears:

$$idf_n = \log \frac{|D|}{|\{n \in d\}|}$$

After the *tf* and *idf* have been calculated for each term in a question, the question's vector *Vd* can now be computed as follows:

$$w_{n,d} = \boldsymbol{f}_{n,d} \cdot idf_n$$

Similarity Function

The similarity function used to measure the similarities between questions considered for mutual inclusion to the test being generated in PARES is based on the convergence of those questions' document vectors. The angle between the vectors is calculated and its cosine is then taken into account. Question pairs where the cosine is equal to 0 are deemed to have no similarity at all, while pairs where the cosine is equal to 1 should be deemed extremely redundant. The following equation highlights the method of similarity calculation:

$$\cos\theta = \frac{V_1 \cdot V_2}{\|V_1\| \cdot \|V_2\|}$$

The vector dot products and magnitudes in the above equation are calculated as follows:

$$V_1 \cdot V_2 = \sum_{n=1}^{N} w_{n,1} \times w_{n,2}$$

$$\|V_1\| \cdot \|V_2\| = \sqrt{\sum_{n=1}^{N} w_{n,1}^2} \times \sqrt{\sum_{n=1}^{N} w_{n,2}^2}$$

Test Generation Process

We can now comprehensively summarize the conflict detection process built into PARES.

There are three distinct phases in the test generation process. In the first phase, preliminary calculations are made after any question is created or edited in order to compute the document vector for all questions. This is necessary as the inverted document frequency for any term may change after any one document is edited, and therefore a change in any document may result in alterations to possibly all document vectors.

```
Phase 1 (question authoring):
Calculate term frequency and inverted document frequency for each term and document.
Calculate document vector for all questions.
```

In the second phase, teachers select the parameters for a test template they wish to make available to students. A key parameter for the test is the number of questions from each teaching unit that should be included in the test; the system must therefore confirm that there are a sufficient number of nonconflicting questions to satisfy this requirement. In order to achieve this, conflicting questions are assigned to a number of bins. It is evident that at most one question from each bin can be used in a conflict-free test; therefore, if the number of bins is smaller than the number of questions to include the test is not viable with the given parameters. PARES also provides support for questions of varying difficulty and creating tests with a specified difficulty level, a feature which we have not addressed in this discussion because it is not related to the conflict detection algorithm. This feature can be implemented by creating sub-bins for each similarity group where questions with differing difficulty are placed.

```
Phase 2 (test creation):
Accept test configuration data from teacher
For each course unit {
        Retrieve S (set of questions pertaining to the unit)
Calculate similarity (vector angle cosine) for each pair of questions in S
Let bin number B = 1
While S is not empty {
Assign any one question Q in S to bin B
Remove Q from S
Assign all questions Q’ with similarity(Q, Q’) > threshold to bin B
Remove all Q’ from S
Let B = B + 1
}
If B – 1 < N (number of questions to be included) {
Not enough material to create test for this N
}
}
```

In the final phase, triggered when a student has elected to take the test, the questions are again assigned to bins as above and a random question is selected from each bin for inclusion until enough questions have been selected.

```
Phase 3 (test generation):
For each course unit {
    Retrieve S (set of questions pertaining to the unit)
    Calculate similarity values (cosines) for each pair of questions in S
    Let bin number B = 1
    While S is not empty {
        Assign any one question Q in S to bin B
        Remove Q from S
        Assign all questions Q' with similarity(Q, Q') > threshold to bin B
        Remove all Q' from S
        Let B = B + 1
    }
    For i = 1 to N (number of questions to be included) {
        Randomly pick a bin P where 1 <= P < B
        Randomly pick a question assigned to B and include it in the test
        Remove bin P from bin list
        Let B = B – 1
    }
}
Present test to student
```

Evaluation

To evaluate the PARES efficiency of finding conflicting questions, 103 exam questions were submitted for three higher education courses: 45 on Telematics, 32 on Distance Education, and 26 on Teaching Information Technology. These questions were either original and had concept dependencies or they were similar to other questions. Since these questions had been previously classified into one of the above subjects, according to their type, the goal of the evaluation was to find the rate of successful question classification per course as well as in total so as to measure the conflict algorithm efficiency. Evaluation results are summarized in Table 1 and Figure 4.

Table 1

Evaluation Results

Telematics				
	Similar	Concept dependencies	Original	Total
Submitted	28	7	10	45
Successfully identified	27	7	10	44
Percentage	96.43%	100.00%	100.00%	97.78%
Distance Education				
	Similar	Concept dependencies	Original	Total
Submitted	20	9	3	32
Successfully identified	18	8	3	29
Percentage	90.00%	88.89%	100.00%	90.63%
Teaching Information Technology				
	Similar	Concept dependencies	Original	Total
Submitted	16	6	4	26
Successfully identified	12	4	4	20
Percentage	75.00%	66.67%	100.00%	76.92%
All Courses				
	Similar	Concept dependencies	Original	Total
Submitted	64	22	17	103
Successfully identified	57	19	17	93
Percentage	89.06%	86.36%	100.00%	90.29%

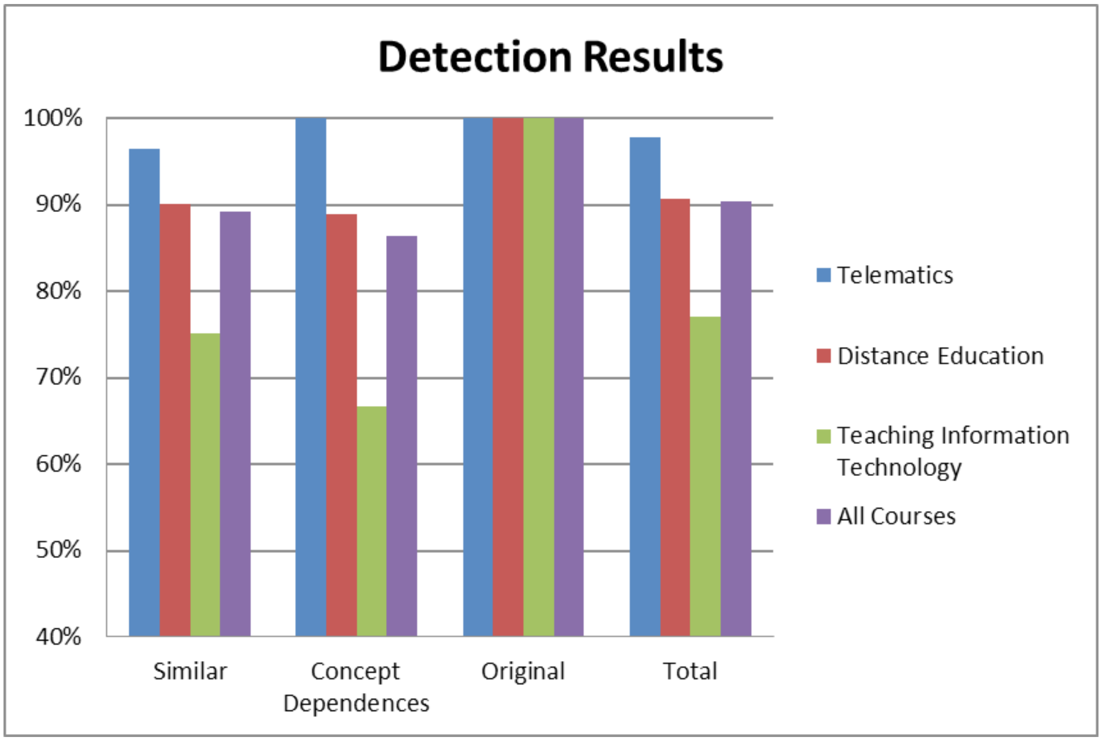


Figure 4. Detection results.

In more detail, out of the 45 questions on Telematics, 28 were similar to other questions, 7 had concept dependencies, and the remaining 10 were original. PARES successfully identified 27 out of the 28 similar questions and all the original and concept dependencies questions. The rate of successful identification in total rose to 97.78%. A small decrease of successful identifications was observed on the Distance Education course questions. Out of the 32 questions submitted, PARES successfully identified 29. Even though there was a decrement in system efficiency, the success rate was still over 90%. This rate, however, appears to decrease significantly in the Teaching Information Technology course questions. More specifically, 20 out of the 26 submitted questions where identified successfully, which amounts to about 76.92%.

In total, PARES successfully identified 93 out of the 103 questions, which corresponds to a success rate of 90.29%. As far as the different types of questions are concerned, PARES succeeded in finding all the original questions, while the success rate for similar questions and those with concept dependencies is at about 89% and 86% respectively.

From the above results, it seems that the conflict detection algorithm that PARES adopts responds with high recognition accuracy to the question types. However, even though there is a very high algorithm success percentage in the Telematics and Distance Education courses, which are courses that make increased use of specific terminology and more questions had been submitted, in the Teaching Information Technology course the success rate is significantly lower. Specifically, in Telematics many standard keywords are used, like ADSL, WiFi, optical fiber, and so on, and proportionally the same applies to the Distance Education course, which contains standard keywords such as distance learning, student model, and so on. It seems, therefore, that the identification of conflicting questions decreases in those cases where the wording of the questions is descriptive and lacks terminology or where limited use of terminology is made. In addition the number of submitted questions in the Telematics course, in which system detection accuracy was the highest, is also at a high level (45) and almost double from the Teaching Information Technology course submitted questions (26). Thus, the number of submitted questions affects the performance of the algorithm.

Fortunately, algorithm efficiency may be further increased through continuing education and the enrichment of the system's correlation vocabulary. This may be achieved either through the submission of more questions related to a particular topic or through the intervention of an expert who correlates the specific keywords and key phrases used in specific topic questions. These actions lead to a higher success rate of the algorithm as there is increased terminology awareness on a particular topic.

Conclusion

Several established e-learning platforms today offer e-testing tools to facilitate evaluation and assessment of the learning process. These tools are still being developed as there are many opportunities for the inclusion of features that will greatly increase the testing material's potential for reuse both in space (by reusing material developed in other platforms or deployments) and time (by combining the same material in different ways for each assessment). These opportunities however present certain problems that must be addressed before such features are ready for productive use.

PARES is an e-testing platform that aims to provide assessment services to academic staff by facilitating the creation and management of question banks and powering the delivery of nondeterministically generated test suites. This capability is very important in cases where teachers wish to provide students with the option of testing their subject knowledge several times during the learning process, a scenario which would require immense amounts of effort if implemented with pre-engineered tests. The platform augments this feature with additional parameters that enable the generation of tests with a specified difficulty level. Therefore PARES may help both teachers and students assess learning performance more efficiently. Consequently this will allow teachers to improve their courses and provide appropriate responses to their students. On the other hand students can readjust their study according to the online tests outcomes.

In order to provide tests that are effective and free of conflicting questions, PARES uses an algorithm based on the vector space model to compute the similarity between questions and exclude questions which are deemed to have an unacceptably large similarity from appearing in the same test suite. Furthermore, teachers can be warned in advance that their question banks are not populated enough to create tests with certain characteristics.

Since the performance of the system depends on its ability to accurately calculate question similarity, further work will naturally focus on improving these calculations. The vector space model used has certain known deficiencies, only some of which may be offset by making larger question banks available. In particular, the use of keywords can be made more effective if they are internally processed to a more computer-friendly form before they are used as input to the algorithm. Stripping words which belong in the rejected keywords list (list of words devoid of specific meaning) from key phrases and stemming keywords (so that grammatical rules do not hinder the operation of the algorithm) are two such obvious improvements, after the implementation of which the term weighting function can be further profiled and improved.

References

Bilenko M, & Mooney RJ. (2003). Adaptive duplicate detection using learnable string similarity measures. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)* (pp. 39-48).

Hage, H., & Aimeru, E. (2006). ICE: A system for identification of conflicts in exams. *Proceedings of IEEE International Conference on Computer Systems and Applications* (pp. 980-987).

IMS GLC (2011). *IMS question & test interoperability specification*. Retrieved from http://www.imsglobal.org/question/

Jiang, H. (2009). Study on the performance measure of information retrieval models. *Proceedings of IEEE International Symposium on Intelligent Ubiquitous Computing and Education* (pp.436-439).

Kaburlasos, V.,G., Marinagi, C.,C., & Tsoukalas, V.,T. (2004). PARES: A software tool for computer-based testing and evaluation used in the Greek higher education system. In Kinshuk et al. (Eds)*, Proceedings of 4th International Conference of Advanced Learning Technologies* (pp.771-773), Joensuu, Finland: IEEE.

Kikusawa, M., Yamakawa, O., & Tanaka, T. (2006). The method and role of CBT in a classroom lecture of higher education. In T. Reeves & S. Yamashita (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 661-666).

Maron, M. E., & Kuhns J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM, 7*(3), 216-244.

Plateau (2011). Plateau Talent Management System. Retrieved from http://www.plateau.com/prod/exams.htm at April 2011

Pritchett, N. (1999). Effective question design. In S. Brown, P. Race, & J. Bull (Eds.), *Computer-assisted assessment in higher education* (pp. 29-37). London: Kogan Page.

Romero C., Ventura S., Hervás C., & De Bra. P. (2005). Extending AHA!: Adding levels, data mining, tests and SCORM to AHA!. *Proceedings of Fifth International Conference Human Systems Learning* (pp. 21-40).

Salton, G., Wong, A., & Yang C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Seet, A.M., & Zualkernan I.A. (2010). An adaptive method for selecting question pools using C4.5. *Proceedings of 10th International Conference of Advanced Learning Technologies* (pp. 86-88), Sousse, Tunisia: IEEE.

Triantafillou, E., Georgiadou, E., & Economides, A. A. (2008). The design and evaluation of a computerized adaptive test on mobile devices. *Computers & Education*, 50(4), 1319-1330.