Using Information Retrieval to Detect Conflicting Questions

Hicham Hage and Esma Aïmeur

Department of Computer Science and Operational Research
University of Montreal

{hagehich, aimeur}@iro.umontreal.ca

Abstract

Although E-learning has advanced considerably in the last decade, some of its aspects, such as E-testing, are still in the development phase. Authoring tools and test banks for E-tests are becoming an integral and indispensable part of E-learning platforms and with the implementation of E-learning standards, such as IMS QTI, E-testing material can be easily shared and reused across various platforms. With the knowledge available for reuse and exam automation comes a new challenge: making sure that created exams are free of conflicts. A Conflict exists in an exam if at least two questions within that exam are redundant in content, and/or if at least one question reveals the answer to another question within the same exam. In this paper we propose using Information Retrieval techniques to detect conflicts within an exam. Our solution, ICE (Identification of Conflicts in Exams), is based on the vector space model relying on tf-idf weighing and the cosine function to calculate similarity. ICE also combines the hybrid recommendation techniques of the EQRS (Exam Question Recommender System) in order to propose replacements for conflicting questions.

Edizioni Erickson – Trento
www.erickson.it

Je-LKS Journal of e-Learning and Knowledge Society — *Vol. 2, n. 1, marzo 2006 (pp. 111-121)*

1. Introduction

E-learning has advanced considerably in the last years. Today, there exist many E-learning platforms, commercial (*WebCT*[1], *Blackboard*[2]) or open source (*ATutor*[3]), which offer many tools and functionalities, some aimed towards teachers and developers, and others aimed towards students and learners (Gaudiosi, Boticario, 2003). Nonetheless, some of E-learning's aspects, such as *E-testing*, are still in their early stages. E-learning platforms offer E-testing Authoring tools and Test Banks, nevertheless, most of these tools are limited to the platform itself and to the best of our knowledge, Test Banks are *limited* to the teacher's private use. With E-learning standards and specifications, such as the *IMS QTI*[4] (IMS Question and Test Interoperability), teachers can *explicitly* share E-testing material by using import/export functionalities, available only on some platforms. In order to encourage knowledge sharing and reuse, we are currently in the works of designing and implementing a web-based *assessment authoring tool* called *Cadmus*. Cadmus offers an IMS QTI-compliant centralized questions-and-exams repository for teachers to store and share *implicitly* E-testing knowledge and resources. Moreover, Cadmus offers tools such as the EQRS (Exam Questions Recommender System) (Hage, Aïmeur, 2005) to help locate required information. Nevertheless, selecting questions depending on the teacher's preference cannot guarantee a flawless exam with no *conflicts*. A conflict exists in an exam if two or more questions are redundant in content, and/or if a certain question reveals the answer of another question within the same exam. Such conflicts might be frequent typically when a teacher is using shared questions, and especially in the automation of the exam creation process. This paper introduces ICE (Identification of Conflicts in Exams), a module within Cadmus that uses IR (Information Retrieval) techniques to identify conflicts within an exam. ICE is based on the *vector space* model using the *cosine* function and *tf-idf* weighing technique (Singhal 2001). Furthermore, ICE combines the EQRS techniques in order to recommend replacements for conflicting questions. The paper is organized as follows: section 2 introduces E-learning, and E-testing; section 3 presents Cadmus; section 4 describes the approach of ICE; section 5 highlights the testing procedure and results; and section 6 concludes the paper and presents the future works.

2. E-Learning

E-learning is the delivery and support of educational and training material using computers. E-learning is an aspect of distant learning, where teaching ma-

[1] http://www.webct.com/.
[2] http://www.blackboard.com/.
[3] http://www.atutor.ca/.
[4] http://www.imsproject.org/.

terial is accessed through electronic media and where teachers and students can communicate electronically. E-learning is very convenient and portable, and involves a great collaboration and interaction between students and tutors or specialists. There are four parts in the life cycle of E-learning: Skill Analysis, Material Development, Learning Activity and Evaluation/Assessment.

2.1 E-testing

E-testing is the development, delivery and support of testing and assessment material using computers. Research done on 908 volunteers from 25 different classes at Ball State University (Butler 2003) indicates that student taking exams on computers have a positive attitude towards a higher number of exams, and that E-testing promotes a higher sense of control within the students and less anxiety about taking exams. There exist many E-learning platforms that offer different functionalities, most offer only basic testing functionalities, and are limited to the platform itself.[5] Furthermore, each E-learning platform chooses a different platform/operating system, its own unique authoring tools, and stores the information in its own format. Therefore, in order to reuse E-learning material developed on a specific platform, one must change considerably that material or recreate it using the target platform authoring tools. Standards and specifications help simplify the development, use and reuse of E-learning material (Mohan, Greer, 2003).

Currently, there are many organizations developing different standards for E-learning, each promoting its own standards.[6] IMS QTI sets a list of specifications used in order to exchange assessment information, such as questions, tests, and results. QTI allows assessment systems to store their data in their own format, and provides a mean to import and export that data, in the QTI format, between various assessment systems.

3. Cadmus

Cadmus offers an IMS QTI-compliant centralized questions-and-exams repository for teachers to store and share E-testing knowledge and resources. A teacher using Cadmus may create his own questions using the Question Authoring Environment (Figure 1), has the choice to keep these questions *private*, or *share* them with other teachers. Furthermore, a teacher can use the Exam Authoring Environment (Figure 1) to access his questions, or shared questions from other teachers, in order to create exams. One of the Exam Authoring Environment functionalities is the EQRS (Exam Question Recommender System), a recommender system to help

[5] http://www.edutools.info/index.jsp.
[6] http://workshops.eduworks.com/standards/.

teachers in their search for exam questions. In order to create a proper exam, one must make sure there are no *conflicts* between the various questions of that exam. A conflict exists between two questions in the same exam if one question reveals the answer to the other, and/or if the two questions are redundant. Such conflicts between questions within the same exam *might* be frequent, particularly when a teacher is using questions authored by others, and especially in the automation of the exam creation process. ICE (Identification of Conflicts in Exams) is a new module imbedded into the Exam Authoring Environment; ICE uses IR (Information Retrieval) techniques, to detect conflicts between questions within the same exam. «Information retrieval (IR) deals with the representation, storage, organization of, and access to information items» (Baeza-Yates, Ribeiro-Neto, 1999). The aim of IR is to provide a user with easy access to the information of his interest, estimating the usefulness of a document to the user and ranking them accordingly. IR systems usually assign documents a numeric score, used for ranking purposes. There are several models for this process (Baeza-Yates, Ribeiro-Neto, 1999; Salton, McGill, 1983); some of the most common models in IR are the *vector space model* and the *probabilistic model* (Maron, Kuhns, 1960).

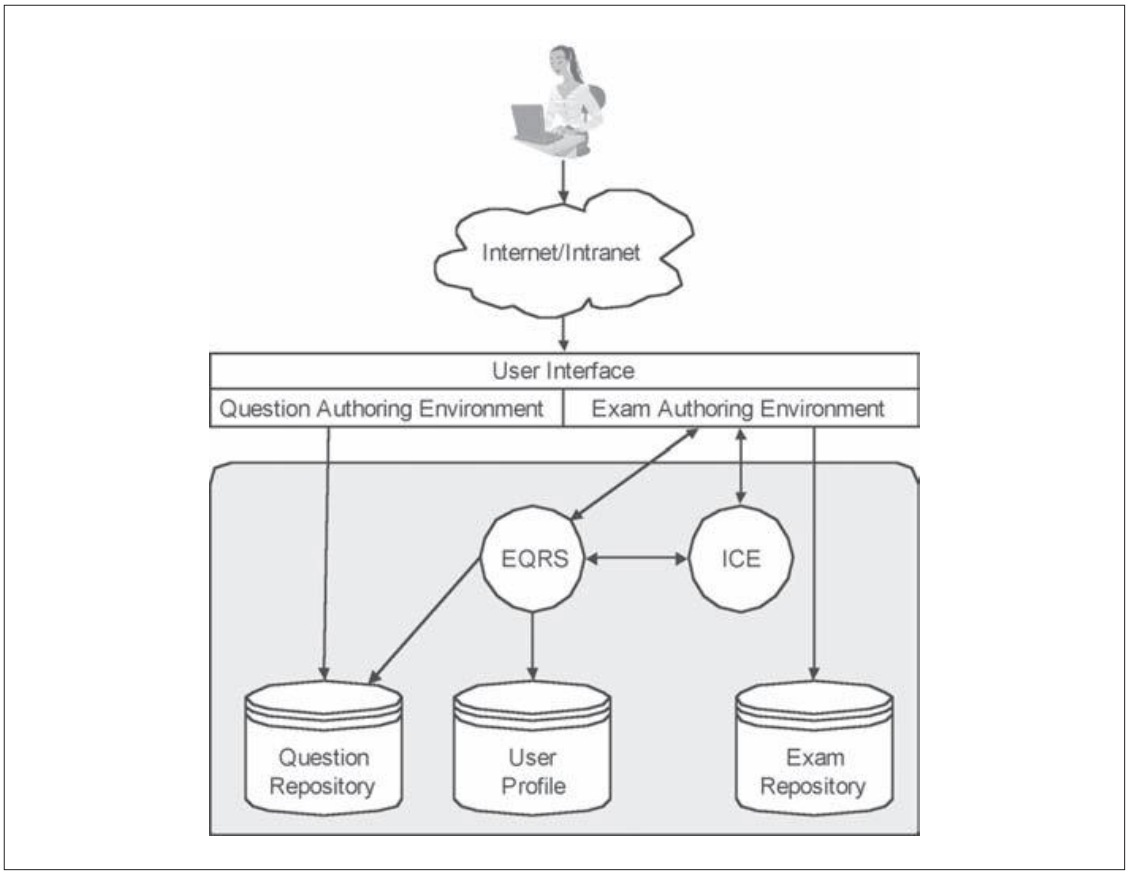

Figure 1 Cadmus Architecture.

4. ICE – Identification of Conflicts in Exams

ICE is a module within Cadmus that detects conflicts between questions within an exam. In order to detect these conflicts, ICE uses IR techniques based on the vector space model. Essentially, the vector space model relies on a *similarity* function to determine how identical the two documents are.

4.1 Similarity Function

In the vector space model, text or a document is represented by a vector of terms (Salton, Wong, Yang, 1975). The Cosine of the angle between two term vectors is used to evaluate the similarity between the respective texts or documents. If the Cosine = 1 then both documents are similar (angle between vectors = 0), and if the Cosine = 0, then the two documents are orthogonal (angle between vectors = 90). Equation 1 highlights the similarity function used to evaluate the similarity (the cosine) between the document vector $\vec{d}_j$ and the query vector $\vec{q}$.

$$\text{sim}(\vec{d}_j, \vec{q}) = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{n} w^2_{i,j}} \times \sqrt{\sum_{i=1}^{n} w^2_{i,q}}}$$

Equation 1: Similarity Function

In Equation 1 $w_{i,j}$ represents the weight of the term i in the document j and $w_{i,q}$ represents the weight of the term i in the query q. In an IR system a query represents what the user is looking for, and the documents represent the search domain. In ICE, the documents are the questions within a specific exam, and the query is one of the exam questions where ICE is trying to determine if any conflicts exist between this *query question* and the rest of the questions within that Exam. When an Author is creating a new question in Cadmus, he is required to specify one or more keywords relating to the content of that question. The terms that compose the query and document vectors are these, author specified, keywords. In order to specify the weight of the keywords ($w_{i,j}$ and $w_{i,q}$) ICE uses the *tf-idf* weighting technique.

4.2 tf-idf weighting

The *tf-idf* weighting scheme relies on the tf (Term Frequency) and idf (Inverted Document Frequency) to determine the weight of a keyword in a certain document. The weight $w_{i,j}$ of a keyword i in a document j is calculated using the formula in Equation 2.

$$w_{ij} = tf_{ij} \times idf_i$$

Equation 2: tf-idf formula

tf_{ij} represents the importance of the term *i* in the document *j*, and is calculated using Equation 3 where $freq_{ij}$ is the frequency of the term *i* in document *j* and max $freq_j$ is the maximum frequency of a term in document *j*. idf_i represents the discriminating power of the term *i* and is determined using the formula in Equation 4, where N is the total number of documents, and n_i is the number of documents in which the term *i* appears in.

$$tf_{ij} = \frac{freq_{ij}}{\max freq_j}$$

Equation 3: tf formula

$$idf_i = Log_2\left(\frac{N}{n_j}\right)$$

Equation 4: idf formula

4.3 ICE Process

Now that the similarity function and keyword weighing scheme is clear, let us put all the building blocks together. Figure 2 illustrates the ICE process. The first step of detecting conflicts in an exam is to select the Exam Questions. Since this process is already completed using the Exam Authoring Environment of Cadmus. There are three stages in the ICE process, preparation (tf-idf calculation), conflict detection, and conflict reporting.

Step1: Preparation
Retrieve Exam data from the Exam Authoring Environment
Set tf = 1
Evaluate idf values for respective keywords
Calculate $w_{ij} = tf_{ij} \times idf_i$

Step2: Conflict Detection
For i = 1 to Number of Questions in Exam {
q ← Question (i)

For j = i+1 to Number of Questions in Exam {
d ← Question (j)
Evaluate S = sim (q_i, d_j)
If S > Threshold then Mark conflict
}
}

Step 3: Conflict Reporting
Report marked conflicts
Offer replacements for conflicting Questions

Figure 2 ICE process.

4.3.1 tf-idf calculation

The first step of the ICE process is to prepare the tf-idf values for the keywords. First, since exam questions are usually short, most keywords will appear only once, thus ICE assumes the tf of all the keywords to be 1. Furthermore, the Exam Authoring Environment keeps track of a counter for each of the various question's keywords; incrementing or decrementing the counter each time a question is added to, or removed from the exam. ICE iterates over the value of the keyword's counter applying Equation 4 to compute the respective idf values. Finally, ICE applies Equation 2 to evaluate the keywords' weights.

4.3.2 Conflict Detection

To detect conflicts within an Exam, ICE iterates the query vector (q_i) on the questions of the Exam, such that i ← 1 to N-1 (N is the total number of questions in the exam). Then, for each q_i, ICE iterates the document vector, d_j, on the remaining questions, where j ← i+1 to N. At each iteration (i,j), ICE calculates S1 = sim(q_i,d_j). If S1 is greater than or equal to the threshold T, then ICE reports Q_i and Q_j as redundant questions. The value of T was determined through testing and is set at 0.45.

Furthermore, at the same iteration (i,j), ICE will automatically extract the keywords of the correct answer(s) of Q_i, and adds these keywords to q_i, resulting in a new query qa_i. ICE then computes S2 = sim(qa_i,d_j). If S2 is greater than or equal to the threshold T, then ICE reports the conflict between Q_i and Q_j: Q_j reveals the answer to Q_i.

Moreover, at the same iteration (i,j), ICE will also automatically extract the keywords of the correct answer(s) of Q_j, then adds these keywords to d_j, resulting in a new document vector da_j. ICE then computes S3 = sim(q_i,da_j). If S3 is greater than or equal to the threshold T, then ICE reports the conflict between Q_i and Q_j: Q_i reveals the answer to Q_j.

4.3.3 Conflict Reporting

When ICE detects a conflict between two questions, that conflict is reported. Both questions are specified with the option to *view* or *replace* each of the questions. To replace a question, the user can search for questions with the same criteria (Type, Difficulty, etc.) as the question to be replaced, or he can change one or more criteria to search for replacement questions.

In the first case, the search for the replacement questions is done through a simple content based filter. All the questions with the same criteria as the question to be replaced are retrieved. ICE will try first to retrieve all the questions with the same criteria as Q_r (the question to be replaced) and none of its keywords. If no

replacement questions were found, ICE will attempt a new search for questions with the same criteria and *some* of Q_r's keywords. In order to know which keywords to allow in the replacement questions, ICE selects Q_r's *prohibited keywords* with the highest weight, such that if a replacement question had all of Q_r's remaining keywords, the similarity will remain less then the threshold T. ICE will perform the new search for all the replacement questions with the same criteria as Q_r and *none* of the prohibited keywords.

In the second case, when one or more search criteria are specified by the user, the search for replacement questions is conducted using the EQRS (Exam Question Recommender System) technique. This approach consists of using a *Feature Combination*, *Hybrid* recommendation technique (Burke, 2002; 2004) to recommend questions.

The recommender system is composed of two levels; a *Content Based* filter and a *Knowledge Based* filter (Burke, 2002). The Content Based filter retrieves a set of candidate questions according to their content, using the same technique for the keywords as described in the previous paragraph. These candidate questions are then sorted by the Knowledge Based filter with regards to their relevance to the user's preferences.

5. Testing and Results

ICE was tested on a questions bank of 200 Java questions. The test generates an exam by selecting between 10 and 40 questions randomly. After the creation of the random exam, ICE will detect the conflicts.

There were a total of 204 randomly created exams with conflicts. The random exams had an average of 28 questions. There were no undetected conflicts, and a total of 512 reported conflicts. Since the same conflict between two questions might appear in several exams, recurring conflicts were grouped into conflict case. Grouping the recurring conflicts into cases resulted in a total of 93 different conflict cases, out of which 77 (83%) were true conflicts and 16 (17%) were not actual conflicts.

These results are illustrated in Figure 3. Most of the invalid conflicts reported are due to keywords selection and weighing. Different questions with very similar keywords, such that the difference in the context of the questions is defined by only one of the keywords, have a similarity greater than the threshold. Increasing the value of the threshold will result in true conflicts being undetected.

Nonetheless, testing proved that setting T to 0.458 (T was 0.45 originally) increased the accuracy of conflict reporting, although now, there are undetected valid conflicts (Figure 4). A further increase in the value of T reduced the number of invalid conflicts reported, but did not ameliorate the accuracy since more true conflicts were passing undetected. Table 1 summarizes the results of the tests.

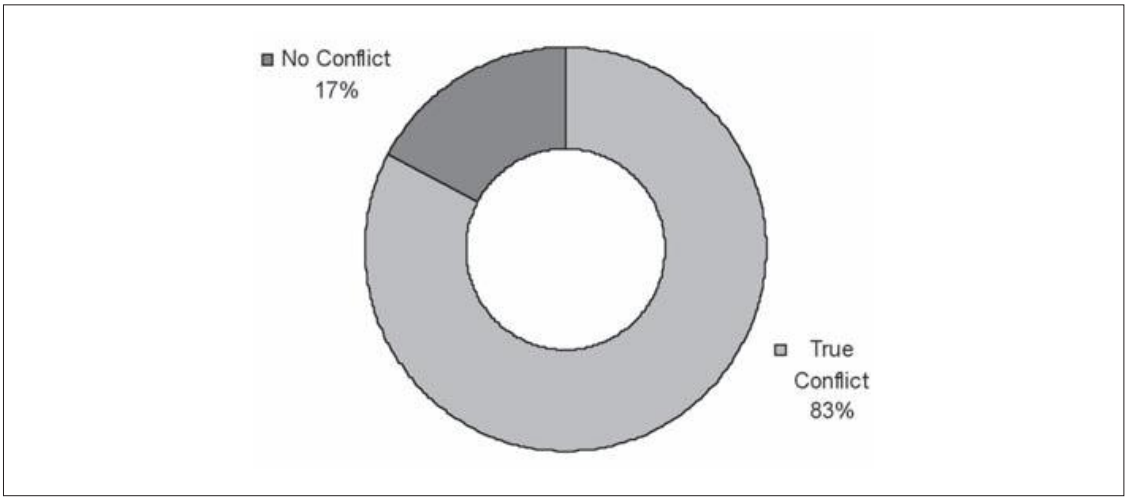

Figure 3 Preliminary Results.

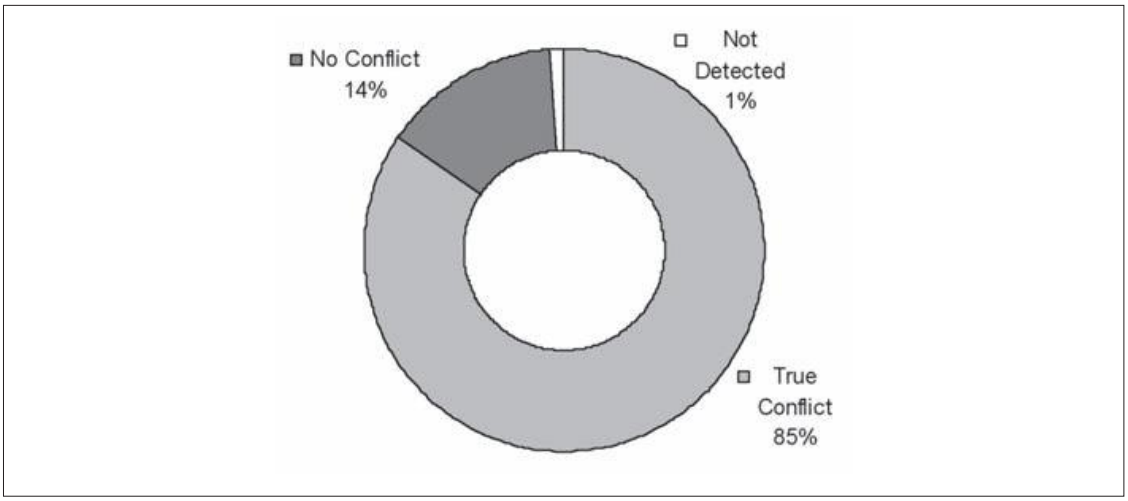

Figure 4 Results after increasing T.

Table 1
RESULTS SUMMARY

	Total	*No Conflict*		*True Conflict*		*Undetected Conflict*	
Preliminary Results	512	111	21.68%	401	78.32%	0	0%
Refined Results	93	16	17.20%	77	82.80%	0	0%
Results T = 0.458	90	13	14.44%	76	84.44%	1	1.11%

Although ICE was tested only on Java questions, the accuracy of conflict detection will not suffer with subjects other than Java since ICE relies mainly on the keywords specified by the author of the question. Initial testing on sample Artificial Intelligence and Databases questions have resulted with a similar, high accuracy conflict detection.

Furthermore, testing on the available question base has revealed that whenever a question Q_i is detected to reveal the answer of a question Q_j, then both questions are similar enough in content to be detected by ICE as redundant questions.

Although it is not a complete surprise (since it is logical to assume that for a certain question to reveal the answer of another question it should be similar in context), further testing on a bigger questions base, and searching for particular cases can help determine the need of testing for such conflicts (if Q_i reveals the answer of Q_j).

6. Conclusion

Today, many E-learning platforms offer E-testing authoring tools. These tools create E-testing material that will remain mostly confined to their author and the platform itself.

Cadmus, an alternative solution, offers an independent IMS QTI-compliant platform to create and share E-testing material. Furthermore, to help the teachers in the exam creation process, Cadmus includes ICE, a module that detects conflicts between questions within an exam. ICE has been tested on a Question Bank of around 200 Java questions.

Results show that ICE conflict detection is quite accurate. After testing ICE on 204 randomly created exams, with an average of 28 questions in each exam, all conflicts were detected, and the accuracy of the conflict reporting was at 83%. Slightly increasing the threshold improved the accuracy by 2%, although some conflicts remained undetected. Furthermore, thus far, testing has shown that whenever a question is detected by ICE to reveal the answer of another question, the two questions are similar enough in content to be reported as redundant. Additional testing, on a larger question bank, is required to decide on the necessity of checking for such conflicts.

The main focus for future work is to enhance the weighing scheme to further refine the accuracy of conflict detection, for instance: taking advantage of the *tf*, such that keywords related to content weigh more that other keywords. Moreover, further consideration on combining tools such as the EQRS (to select the questions) and ICE (to validate the exam) in order to automate the exam question selection, bearing in mind restrictions such as to include (and not exclude due to conflicts) questions to cover the exam domain.

BIBLIOGRAPHY

Baeza-Yates R. & Ribeiro-Neto B. (1999). *Modern Information Retrieval*. Reading, MA: Addison-Wesley.

Burke R. (2002). Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction* 12(4): 331-370.

Burke R. (2004). Hybrid Recommender Systems with Case-Based Components. *Advances in Case-Based Reasoning, 7th European Conference (ECCBR 2004)*, 91-105, Madrid.

Butler D.L. (2003). *The Impact of Computer-Based Testing on Student Attitudes and Behavior*. The Technology Source. URL: http://technologysource.org/article/impact_of_computerbased_testing_on_student_attitudes_and_behavior/ accessed on July 2005.

Gaudiosi E. & Boticario J. (2003). Towards web-based adaptive learning community. *International Conference on Artificial Intelligence in Education (AIED 2003)*, 237-244, Sydney.

Hage H. & Aïmeur E. (2005). Exam Question Recommender System. *International Conference on Artificial Intelligence in Education (AIED 2005)*, 249-257, Amsterdam.

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

Mohan P. & Greer J. (2003). E-learning Specification in the context of Instructional Planning. *International Conference on Artificial Intelligence in Education (AIED 2003)*, 307-314, Sydney.

Salton G. & McGill M. (1983). Introduction to modern information retrieval. New York, NY: McGraw-Hill.

Salton G., Wong A. & Yang C.S. (1975). A vector space model for information retrieval. *Communications of the ACM* 18(11): 613-620.

Singhal A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin*, 24(4): 35-43.