A Collaborative Algorithm for Ontological Matching in E-Learning Courseware Domain Knowledge Network

Akanbi Caleb [1] and Adagunodo Rotimi.[2] Omotosho Lawrence [1] and Adigun Adepeju[1].

[1] Department of Information and Communication Technology, Osun State University, Osogbo, Osun State, 23436, Nigeria

[2] Department of Computer Science and Engineering,Obafemi Awolowo University, Ile-ife Osun State ,Nigeria

Abstract

Domain Knowledge is the content repository of a courseware system consisting of a series of learning objects. However, the unstructured and inconsistent naming of domain knowledge components does not permit knowledge transfer across diverse collaborative systems due to differences in architecture, format and representations. To address this identified problem, we formulate an ontological matching algorithm that provides a sharable knowledge in collaborative learning environment in this paper. The Algorithm employs Hybrid Similarity Measure to compute both Concept and Relational similarity values of the various input graphs-learning objects .

Keywords: Courseware Systems, Domain Knowledge Network, Similarity Measure, Ontological Matching.

1.0 Introduction

e-Learning Courseware System (e-LCS) is an educational software platform for implementing e-learning programmes in tertiary institutions and cooperate organisations. It can organize, present, manage, evaluate the contents of courses and teaching activities, to promote the interaction between students and teachers[1]. e-LCS simulates the traditional learning classroom environment by using syllabus, schedule, course notes, examples, assignments, etc. In addition, it has on-line facilities such as online assessment and multimedia course delivery over the networks which are additional advantages over lecture only classes. Courseware major components are learner's model, on-line facilities such as online assessment, multimedia course delivery tools/environment for collaboration, learners model, facilitators models, pedagogical contents and domain knowledge [2].

Learning object is a smallest digital reproducible and addressable resources of a learning contents stored in various Knowledge Base of Learning Management Systems[3]). Domain Knowledge is a learning materials or content repository of a courseware system. It consists of a series of learning objects arranged in hierarchical format [4]). The knowledge network represents the content to be covered in e-learning system. Usually, it is always in hierarchical, tree or graph forms. The knowledge network represents concepts at different abstraction levels and combines multiple relationships such as *is-a*, *part-of*, *contained-in*, *associate -with*, *related-to*, *example-of*, *applicable-to*, *easier-than*, etc. in a single intuitive hierarchy[5,6,7,8]

The high rate of e-learning platforms implies that increasing complex and dynamic web based infrastructures need to be managed more efficiently [9]. However, the unstructured and inconsistent format of domain knowledge components does not permit knowledge transfer across diverse collaborative systems due to differences in architecture, format and representations. To address this identified problem, we formulate an ontological matching algorithm in e-learning courseware systems that provide a sharable knowledge in collaborative learning environment in this paper.

1.1 Structure of a Domain knowledge

Domain knowledge is composed of other knowledge using some operations called associations or aggregations. Associations are some relationships defined on "simpler knowledge". Two basic categories of knowledge are ; *unit and aggregated* knowledge.). Learner/ Teacher treats it as indivisible item in the given context, unit cannot be decomposed into smaller parts . Aggregated knowledge is composed of either of atomic knowledge units or of other "lower-level" aggregated knowledge. Any kind of knowledge can be grouped and sequenced into clusters. Such a cluster is aggregated knowledge too.

In the view of [10], Learning object (LO) model consists of three basic parts: name, knowledge-based interface and knowledge-based body. The first, such as the topic/theme name, is for identification and referencing or the learning objective statement (e.g., as the context of the name). Interface is for communicating and transferring knowledge to the LO and from it. Some learning objects (knowledge) in that part may be teacher-oriented (e.g., tasks for test, answers, etc.) while the other objects are student-oriented. What is common to all the above mentioned structural units of the model is that within each section the information is strongly clustered into blocks.

As teacher and learner communicate knowledge, interface can be seen also as a media for the teacher/learner interaction, consists of two kinds of knowledge: input and output knowledge. From the learner's perspective, output knowledge is shift in time with respect to input knowledge within a given learning process when it is initiated. The conceptually input knowledge is called 'prerequisites' in pedagogy theories [11]). A typical structure of learning object(LO) model is consists of knowledge-based interface, knowledge-based body, Declarative part, Procedural part, Visibity layers, Contextual part and Managerial part.

1.2 Ontology Matching in Knowledge Network.

The benefits of ontological representation of domain knowledge lie in its capabilities of explicitly defining concepts and their attributes and relationships. Coupled with new information technologies, such representation can be encoded in ways that allow for direct conversion into implementation models. This in turn requires ontological modeling of domain knowledge not only to cover the content but also to take into consideration of how the content is to be used and interacted with users or other courseware systems . Ontological modeling for learning objects may be divided into three broad areas: content, presentation, and application.

Matching is a critical operation in many application domains, such as semantic web, schema/ontology integration, data warehouses, e-commerce, query, mediation, etc. It takes as input, two schemas/ontologies, each consisting of a set of discrete entities (e.g., tables, XML elements, classes, properties, rules, predicates) and determines as output the relationships (e.g.,equivalence, subsumption) holding between these entities [12]

According to Shvaiko (2004), A *matching element* is a 5-tuple:(*id, e , n, R, P(t, s))* where

id - is a unique identifier of the given mapping element;

e - are the entities (XML elements) of the first ontology $e^{/}$ and the second ontology $e^{//}$ respectively.

n - is a *confidence measure* in some mathematical structure typically in the [0,1] range holding for the correspondence between the entities $e^{/}$ and $e^{//}$

R - is a relation (e.g., *equivalence*; *disjointness*; *overlapping)* holding between the entities *e* and $e^{/.}$

r - is an *alignment matching* parameter which is an external resource(s) used by the matching process to extend the definition of the matching process(synonyms).

p- represents the matching parameters (such as weights and thresholds)

The match operation is defined as a function that takes two schemas S_1 and S_2 as input graph and returns a mapping between those two schemas as output. Figure 2 shows the match operation where *p* represents the matching parameters (such as weights and thresholds) and *r* represents the external resources used by the matching process, e.g. thesauri and synonymies, etc.

1.3 Graph Similarity in a Knowledge Network

Graphs are widely used to represent complex structures that are difficult to model. In a labeled graph, vertices and edges are associated with attributes, called labels. In a Knowledge Network, the attributes could be tags in XML documents, atoms and bonds in chemical compounds, genes in biological networks, and object descriptors in images. Using labeled graphs or unlabeled graphs depends on the application need [13],[14].

The vertex set of a graph G is denoted by V(G) and the edge set by E(G). A label function, *l* maps a vertex or an edge to a label. The size of a graph is defined by the number of edges it has, written as |G|. A graph G is a subgraph of G′ if there exists a subgraph isomorphism from G to G′, denoted by G c G′. G′ is called a super graph of G. Given a graph database and a query graph, we may not find a graph (or a few graphs) in the database that contains the whole query graph. Thus, it would be interesting to find graphs that contain the query graph approximately, which is a substructure similarity search problem.

Definition 1: Substructure similarity search.

Given a graph database D = {G1,G2, . . . ,Gn} and a query graph Q, similarity search is to discover all the graphs that approximately contain query graph Q. Reverse similarity search is to discover all the graphs that are approximately contained by this query graph. To distinguish a query graph from the graphs in a database, we call the latter, target graphs.

The structure-based similarity measure directly compares the topology of two graphs, However, since this measure takes structure connectivity fully into consideration, it is more accurate than the feature-based measure [15].

2.0 Literature Review

An ontology according to [14] is an explicit formal specification of a conceptualization and abstract and simplified vision of the world to be represented. Thus, an ontology permits the capturing of knowledge regarding a concrete domain.

Many ontological structure methods have been employed in the courseware domain knowledge modelling. [4] proposed four layer of educational meta model ontology consisting of four interconnected model types in a hierarchical form. These are: Course layer, Module layer, Concept layer (structured as conceptual network), Instruction Microfunction layer. Matching approaches could be Element matching, Structure matching, Hybrid matching approach combines different approaches together, using a hybrid matcher or a composite matcher. A hybrid matcher integrates several approaches based on multiple matching criteria. while a composite matcher combines multiple match results produced by different matchers, including hybrid matchers

A number of graph similarity algorithms have been proposed in the literature [15], [16] These are:

(i) Three Edit Distance Algorithm which finds the similarity of two trees by computing the minimum total number of edit operations to transform one tree into the other by applying three elementary operations, namely insertion, deletion and relabeling However, the tree edit distance algorithm has the problem of a “scattering effect” and mainly focuses on node-labeled.

(ii) The Weighted Tree Similarity Algorithm computes the similarity between node-labeled, arc- labeled and arc-weighted trees. The algorithm can be applied to various environments where weighted trees are used. The algorithm calculates the edit distances between Undirected Acyclic Graphs (UAGs), which is a natural extension of the edit distance for strings and trees.

(iii) Similarity Flooding Algorithm Similarity Flooding Algorithm is a graph matching algorithm in which the input schemas are converted into directed acyclic labeled graphs. The algorithm uses iterative fix point computations called *similarity measures* to determine the similarity of nodes, relying on the intuition that whenever any two elements are found to be similar, the similarity of their adjacent elements increases [4]

During a match operation, similarity measures are used to determine the level of similarity of ontologies O_1 with O_2 .In the view of [16]), a similarity measure for comparison of ontology entities could be object equality, explicit equality, string equality and dice coefficient [17].

3.0. MACA Collaboration Algorithm

The objective of this paper is to formulate a collaboration algorithm for the mediating agent architecture (MACA). The interaction protocol between the requesting courseware, provider courseware and MACA in a collaborative environment is represented by the UML sequence diagram in figure 3.9.

The Multi- Agent System Architecture is shown below consisting of Courseware Mediator Interphase(CMI), Collaboration Agent Wrapper Module (CAWM), Collaboration Agent Match Module(CAMM), Recommender Engine. The details has been discussed in [18]. The architecture consists of two main processes- Match and Filtering processes.

The System Architecture adopts the following protocol interaction.

(i) Filter agent receives ALQ and creates collaboration agent to execute

(ii) MACA collaboration agent migrates to other courseware in the collaborative environment, to search for similar learning objects (LOj) one after the other.

(iii) Collaboration Agent Wrapper Module (CAWM) generates the XML. file format for the active learner query file (ALQ_i), and the learning object files $LO_{J(1......n)}$ found in P-courseware.

(iv) Collaboration Agent Match Module(CAMM) computes the similarity values(MatchSimVal) between ALQ_i and $LO_{J(1......n)}$ within *confidence measure* of interval[0,1] in order to determine the degree of similarity. This is achieved by the Hybrid Graph Similarity Flooding Algorithm (HGSFA) discussed in section

(v) Filter agent searches for similar cases within the knowledge base Recommended Learning Object Repository (RLOR) that matches the ALQ and stores them and their similarity value in MLOR

(vi) filter agent receives the computed learning objects $LO_{J(1......n)}$ and their computed MatchSimVal returned by the collaboration agent and stores them in Match learning Object Repository (MLOR).

(vii) Filter agent computes and recommends to the learners, learning objects with 5-top similarity values. This is achieved by:
- generating a ranked list of returned $LOj_{(1...m)}$ based on their MatchSimVal using sort algorithm.
- selecting the ($LO_{J(1......n)}$ with MatchSimVal >= threshold. In this study the researcher set the threshold at 0.5.

(viii) Active Learner can select any of $LO_{j(1...m)}$ he wishes to study out of the maximum 5-top learning objects recommended .

(ix) Collaboration process is established between the R-courseware and the P- courseware with the selected similar learning object.

3.1 Ontological Similarity Model

The graph similarity flooding scheme is employed in this study to model the proposed collaboration algorithm. The algorithm compares both the concept and structural matching using a dice coefficient similarity measure. The measures for similarity computation can be divided into two generic groups namely *Concept/Lexical Measures* and *Relational/Structural Measures (*Vargas-Vera *et al.*, 2004). Lexical measures are based on surface similarities such as that of the title, label of entities(learning objects). The main idea in using such measures is that usually, similar learning objects have similar names and descriptions across different ontologies.

On the other hand, structural measures try to recognize similarities by considering the kinship of the components and structures residing in the ontology graphs of the knowledge base. Leveraging other available information in two ontologies, they hope to recognize related entities outside the site of the lexical measures. In this study similarities our algorithm rely on the intuition that *elements of two distinct models are similar when their adjacent elements are similar.*

In this study, the collaboration algorithm proposed(called MACA_ Algorithm) employs hybrid graph similarity flooding algorithm(HGSFA) equation in 3.1 to compute both *Concept* and *Relational values* for the input graphs. HGSFA builds the graph for the entities (ALQ and P-courseware learning object) and compute both concept and relational similarities (all in XML.format).

For concept similarity, if the nodes are the same, it is represented as 1, if the nodes are not, it is represented as 0. The relational similarity is determined by dice coefficient equation in 3.2 .

$$\text{Dice Coefficient} = \frac{2*\Sigma \ln a_i b_i}{(\Sigma \ln a_i^2 + \Sigma \ln b_i^2)} \quad (1)$$

where vectors a_i and b_i are the numbers of concept nodes of graph G1 and G2 respectively. The model for the Hybrid Graph Similarity Flooding Algorithm HGSFA is represented as;

$$H = \begin{cases} 0 & \text{no common concepts} \\ 1 & \text{same set of concepts, otherwise} \\ sim\_dice(A,B) = \dfrac{2*\Sigma \ln a_i b_i}{(\Sigma \ln a_i^2 + \Sigma \ln b_i^2)} & \end{cases} \quad .... (2)$$

The similarity value is therefore the average of the computed concept similarity value and relational similarity value. i.e. MACA_Similarity Value = (Concept _SimVal + Relational_ SimVal)/2 …… …(3)

3.3 System Algorithm

The focus of this paper is to present the algorithm of the collaboration algorithm for the architecture. The collaboration algorithm uses Hybrid Graph Similarity Flooding(HGSF) scheme that employs Dice Coefficient Similarity Measures. This Hybrid algorithm has two main procedures – Filtering and Match. The two are used for ontological matching of domain knowledge network. The input and output automaton of the proposed collaboration algorithm is shown in figure 3

4.0 Conclusion

One of the goals of collaboration in e-learning system is to provide sharing of learning objects across diverse courseware domain knowledge. This paper proposed an ontological matching algorithm for learning object sharability across collaborative courseware e-learning objects. The algorithm employed a dice coefficient measure to compute the Concept and Relational values of the objects

References

[1] J.H. Li, Y. Zhao, On the Course Management System (CMS), Modern Educational Technology, 9(18) (2008) 64-71

[2] Monari M.(2005): Evaluation of Collaborative Tools in Web-Based E-Learning Systems M.Sc thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology Stockholm, Sweden.

[3] E.J.Koper, E.J.(2003).: Combining re- usable learning resources and services to pedagogical purposeful units of learning. In A.Littlejohn (Ed.), Reusing Online Resources: A Sustainable Approach to eLearning London Kogan Pgs. 46–59

[4] Kanongchaiyos P.(2007): Calculation of Document Similarity using Cellular Structured Space Template. Proceedings of the third/ASTED International Conference Advance in Computer Science and Technology, Phuket, Thailand.

[5] P. Fournier-Viger, A. Mayers, M. Najjar, R. Nkambou. A Cognitive and Logic Based Model for Building Glass-Box Learning Objects. Interdiscipli-nary Journal of Knowledge and Learning Objects, Vol. 2, 2006, 77-94.

[56] V. Ponkratius. Aspect-Oriented Learning Objects. Proc. of the IASTED International Conference on WEB-BASED EDUCATION, February, Grindelwald, Switzerland, 2005, 21-23.

[7] A. Berlanga, F.J. Garcia. A Proposal to Define Adaptive Learning Designs. Proceedings of Workshop on Applications of Semantic Web Technologies for EducationalAdaptive Hypermedia (SW-EL 2004) in AH 2004, 23 August, 2004, Eindhoven, The Nether-lands. TUE Computer Science-Reports 04-19 AH2004: Workshop Proceedings Part II.

[8] Dolog P. Hese N., Nejdl W. and Sintek M., (2004): Personalisation in distributed e-learning Environments . In Proceeding of the 13th International Word Wide Web Confrence New York, USA.

[9] Štuikys V. and Damaševičius R. (2007): Towards Knowledge- Based GenerativeLearning Objects Learning Objects, Information Technology and Control Vol. 36 No 2

[10] Merrill M.D.(2000) **:** Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects. C.M. Reigeluth (Ed.), Instructional-Design Theories and Models: A New Paradigm of Instructional Theory, Mahwah, NJ: Lawrence Erlbaum Associates

[11] Raymond J., Gardiner E. and Rasca P. (2002): Calculation of graph similarity using maximum common edge subgraphs. The Computer Journal, Vol 45:pp 631–644.

[12]Kalfoglou Y. and Schorlemmer M.(2003): Ontology Mapping: The State of The Art. The Knowledge Engineering Review Journal (KER), Vol18 No. 1, pp1–31.

[13] Shasha D. Wang J, and Giugno R. (2002): Algorithmics and applications of tree and graph searching. In Proc. 21st ACM Symposium on Principles of Database Systems (PODS'02), pages 39–52.

[14] Melnik, S., Molina, H.G. and Rahm, E.(2002): Similarity flooding: A versatile graph matching algorithm, Proceedings of Eighteenth International Conference on Data engineering San Jose, California, pp. 117-128.

[15] Bhavsar, V.C. Boley, H. and Yang, L.(2004): A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments, Computational Intelligence, Vol 20 No 4, pp.584-602.

[16] Qin, J. & Finneran, C. (2002). Ontologicalrepresentation of learning objects. In Proceedings of the Workshop on Document Search Interface Design and Intelligent Access in Large-Scale Collections, July 18, Portland, OR. http://xtasy.slis.indiana.edu/jcdlui/uiws.html

[17] C.O. Akanbi C.O., E.R. Adagunodo. and .O.Omotosho.(2009) : An Intelligent model for Web-based learning management systems International Journal of computer science and Issues Vol 8 issues 4

[18]Vargas-Vera M. and Motta E. (2004):AQUOntology-based Question Answering System. Third International Mexican Conference on Artificial Intelligence (MICAI-2004), LNC2972, Springer Verlag.

Authors

First Authors : *Dr. Akanbi C. Olufisoye.*
Department of Information and Communication Technology, Osun State University, Osogbo is a lecturer I in the above named Dept. He obtained his first degree (B. Sc. Mathematics) in 1991, from University of Ilorin , He holds M.Sc. and Ph.D degrees in Computer Science in 1999 and 2010 respectively from Obafemi Awolowo University, Ile-Ife. He is the Acting Director of Information Management and Technology Centre Osun State University Osogbo. His area of specialization is

electronic learning (e-learning), collaborative Learning , Multi-Agent System, and Artificial Intelligence .

Second Author : Prof. Adagunodo E. Rotimi.
Department of Computer Science and Engineering ,Obafemi Awolowo University, Ile-ife is a Prof. of computer science. He obtained his B.Sc, M.Sc and Ph.D in Computer Science from the same University. He is currently the Head of department of Computer Science and Engineering, He has supervised many M.Sc and Ph.D students and published in many international and local journals. Prof. Adagunodo area of research works include operating system, e-learning and soft computing.

Third Author Mr Omotosho Lawrence O
Mr Omotosho is a Lecturer in the department of Information and Communication Technology, Osun State University, Osogbo, Nigeria. *He obtained his first degree (B. Agric. Soil Science) in 1990, PGD and M.Sc. in Computer Science in 1995 and 2001 respectively from Obafemi Awolowo University, Ile-Ife. He is currently a PhD student in the Department of Computer Science & Engineering, Obafemi Awolowo University, Ile-Ife. His areas of specializations are Health Informatics, Mobile Computing and Software Engineering*.

Fourth Author: Mrs. Adigun Adepeju, A.
Department of Information and Communication Technology, Osun State University, Osogbo is a lecturer in the above named department. She obtained the degrees of B. Sc. (Computer Science) from University of Ilorin, M. Sc. (Computer Science) and M.Pil (Computer Science) both from University of Ibadan. Her area of specialization is Human Computer Interaction (HCI), Operating System (OS) and Collaborative system.

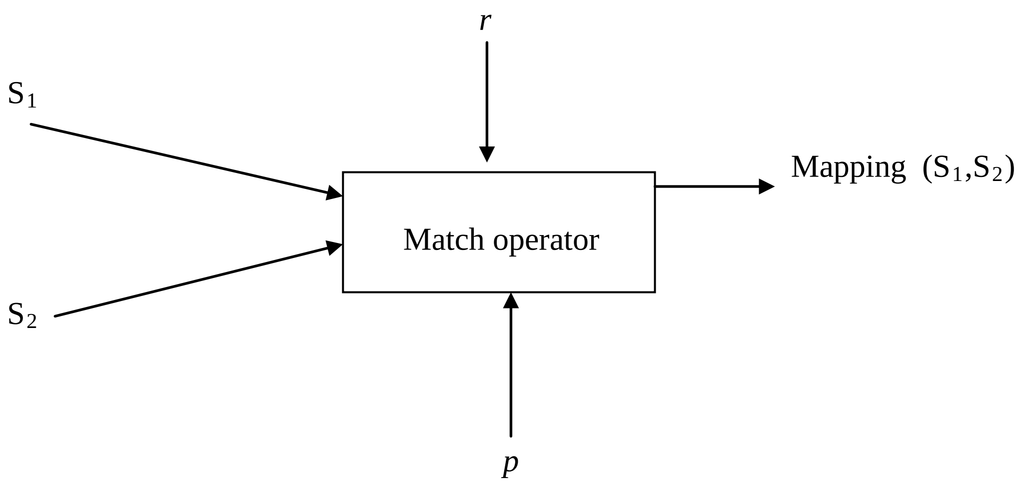

Figure 2: The Match Operation

Input ((Req_Mediating_ Agent-CMI (reqCourswID, ALQ)) Accept learner Query

Precondition:

Effect: delQ: append(delQ (reqCourswID, ALQ))

Search (Simcase(RMOR, ALQ))

Precondition:

Effect: SearchSimCase = append (MLOR,(MatchCase,MatchSimVal)

Deploy((Collab_agent(ProvCourswID, delQ))) Collaboration Agent migration

Precondition:

Effect: SearchProvCoursw = append(ProvCourswID, reqCourswID, ALQ)

Generate(ProvCoursew(Wrapper(delQ.Xml, ProvCoursewKnowlBaseOntol.Xml)))

Conversion to XML formats

Precondition:

Effect:WdelQ:Append(delQ.Xml, ProvKnowlBaseOntol.Xml)

Compute(MatchSimVal(WdelQ)$_i$) for all ProvCoursw$_i$ (i=1 to n) visited

Precondition: Call (**MACA_ Similarity Algorithm**)

Effect: T_ MROR$_i$ = Append (MatchSimAlgorithm(WdelQ)$_i$)

Send{MLOR (T_MLOR)} *for all ProvCoursw$_i$ (i=1 to n)*

Precondition

Effect: MLOR: Append (T_ MLOR, ProvCourswID)$_i$

Perform (Filtering _ Algorithm(Sort(MLOR)

Precondition: Call (***Procedure Filtering & Procedure Quick sort***)

Effect:RLOR =append(Filtering_Algorithm(SortAlgorithm(MLOR))

Send {(result (Max_5-top(RLOR),ReqCoursew)

Precondition:

Effect: RecQ=Append 5-top(RLOR, ProvCourswID (ReqCoursew)

Figure 3: Collaborative Algorithm for Ontological Matching

Procedure MACA_Similarity Algorithm;

Compute Concept _SimVal:

$$\text{Concept\_similarity} = \begin{cases} 0 & \text{no common concepts} \\ 1 & \text{same set of concepts , otherwise} \end{cases}$$

Where vectors A and B are filled with the number of concept nodes of graph G1 and G2 respectively.

Compute Relational_ SimVal:

Ontol_ similarity = di = sim_dice(A,B) = $2*\Sigma \ln a_i b_i / \Sigma \ln a_i^2 + \Sigma \ln b_i^2$

If ontol_relation $\neq 0$
Then evaluate_query (ontol_relation(β1,β2))
else
Obtain synonyms for L_query using FODOC thesaurus
Ask active learner to select choice from FODOC thesaurus provided
Call evaluate_queryl(selected_sense(β1, β2))

Compute MACA_SimilarityVal = (Concept _SimVal + Relational_SimVal)/2

Procedure Filtering

Input all Match_SimVal

Rank Match_SimVal perform quick sort algorithm

Threshold_Val = 0.5

Display all Match_SimVal>= Thresold_Val

Recommend maximum of 5-top ranked Match_SimVal, learning objects and the collaborative courseware for the learner

If 5-top ranked Match_SimVal, learning objects not found threshold Val refined to 0.2 and list learning objects