



HAL
open science

Semi-Supervised Algorithm with Knowledge-based Features for Learner's Profiles Interoperability

Leïla Ghorbel, Corinne Amel Zayani, Ikram Amous, Florence Sèdes

► **To cite this version:**

Leïla Ghorbel, Corinne Amel Zayani, Ikram Amous, Florence Sèdes. Semi-Supervised Algorithm with Knowledge-based Features for Learner's Profiles Interoperability. *International journal of technology enhanced learning*, 2018, 10 (1/2), pp.137-159. 10.1504/IJTEL.2018.088343 . hal-02562112

HAL Id: hal-02562112

<https://hal.science/hal-02562112>

Submitted on 4 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <https://oatao.univ-toulouse.fr/22086>

Official URL

<https://www.doi.org/10.1504/IJTEL.2018.088343>

To cite this version:

Ghorbel, Leïla and Zayani, Corinne Amel and Amous, Ikram and Sèdes, Florence *Semi-Supervised Algorithm with Knowledge-based Features for Learner's Profiles Interoperability*. (2018) International Journal of Technology Enhanced Learning, 10 (1/2). 137-159. ISSN 1753-5255

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Semi-supervised algorithm with knowledge-based features for learner's profiles interoperability

Leila Ghorbel*, Corinne Amel Zayani and
Ikram Amous

Department of Computer Science,
MIRACL-ISIMS Sfax University,
Tunis Road Km 10,
3021 Sfax, Tunisia
Email: leila.ghorbel@gmail.com
Email: corinne.zayani@isecs.rnu.tn
Email: ikram.amous@isecs.rnu.tn
*Corresponding author

Florence Sèdes

Department of Computer Science,
Paul Sabatier University, IRIT Laboratory, Toulouse, France
Email: sedes@irit.fr

Abstract: Nowadays, the user can have several profiles found in different adaptive systems relative to various fields. In particular, adaptive e-learning systems respond to a strong need to adapt to each learner their proposed activities based on the data stored in his/her profile (learning-style, interest, etc.). However, each system can have incomplete data as far as the learner is concerned. Hence, the exchange of the learner's profile data is extremely important in order to enhance his/her learning experience. The exchange requires a matching process so as to resolve the large number of a learner's profiles differences whether in syntax, structure or semantics. In this context, we propose a matching process to automatically detect the similarity between the profile elements. The originality of this process resides in the fact that it rests on a new semi-supervised Tri-Training algorithm which significantly improves the state of the art approaches.

Keywords: e-learning; interoperability; knowledge-based features; learner's profiles; mapping; matching; Tri-Training.

Reference to this paper should be made as follows: Ghorbel, L., Zayani, C.A., Amous, I. and Sèdes, F. (2018) 'Semi-supervised algorithm with knowledge-based features for learner's profiles interoperability', *Int. J. Technology Enhanced Learning*, Vol. 10, Nos. 1/2, pp.137–159.

Biographical notes: Leila Ghorbel received her master's degree in computer science and multimedia from the Higher School of Computer Science & Multimedia of Sfax (Sfax University, Tunisia) in 2013. She is currently a PhD Student from the Faculty of Economics and Management of Sfax since 2014. She is also a member of MIRACL laboratory of Sfax University, Tunisia. Her current research areas include adaptive systems, personalization, user profile, e-learning, social network, interoperability, data mining.

Corinne Amel Zayani is an associate professor in Computer Science at the Faculty of Sciences in the University of Sfax in Tunisia. She received the master's degree in Computer Science from University of Valenciennes and Hainaut-Cambresis in 2001. She got her Ph.D from University Paul Sabatier at Toulouse 3 in 2008. She is a member of MIRACL laboratory of Sfax University. Its current research areas include distributed and social adaptive systems, semi-structured documents, e-learning, interoperability, and Web services.

Ikram Amous received her master's degree in data processing from Paul Sabatier University (Toulouse III, France), France, in 1999. She obtained her PhD in informatics from the Paul Sabatier University in December 2002. She is currently an associate professor at the Higher school of Electronic and Telecommunication of Sfax in Tunisia (Enet'Com). She is also a member of MIRACL laboratory of Sfax University, Tunisia. Her research interests include social network and user profile analysis, adaptive semi-structured document, etc. She has participated in several program committees of national and international conferences.

Florence Sèdes is full Professor in Computer Science at the University Toulouse 3. She got her PhD degree in 1987. From 2011 to 2013, she was appointed as the Deputy Director of the Toulouse Research Lab in Computer Science (Institut de Recherche en Informatique de Toulouse, IRIT), a joint unit with the National Scientific Research Center (CNRS). She led the national research network i3 (Information, Intelligence, Interaction). She is a member of the National University Council of the French Ministry. Her research areas concern data management and information systems. She leads international projects about web services, documents, multimedia metadata and data security. Applications in IoT, CCTV, forensic and social interactions illustrate her contribution on social media ecosystems.

1 Introduction

The learner's profile is a key element in the adaptive educative systems (Brusilovsky and Millan, 2007). It contains several data, such as the learned courses and activities, preferences, interests, etc. These data are represented by different standards, such as Public And Private Information for learner¹ (PAPI), Information Message Service Learner Information Package² (IMS LIP) and Information Message Service Reusable Definition of Competency or Educational Objective² (IMS RDCEO).

A learner's profile can be either empty or contain very little information with incomplete or partial data. Therefore, no learner adaptation can be fulfilled. As a consequence, there is a strong need to exchange, among the different systems, the profile data of the same learner to enhance and enrich his/her profiles. The learner's profile data exchange from one system to another (standard to standard) requires an implementation of a learner's profiles interoperability system. This system aims at adapting the learners' access to the proposed activities grounded on their learning experience (Walsh et al., 2013; Martínez-Villaseñor et al., 2014). It provides mainly a common structure, semantics and syntax for learner's profiles by applying a mapping rule set which is generated by a matching process.

Each mapping rule expresses the similarity between a learner's profile element and another element of a further learner's profile (word pair composed of two element names). Several studies in different domains have been set forward to solve the problems related to the matching process (Martínez-Villaseñor and González-Mendoza, 2017; Zghal et al., 2015; Martínez-Villaseñor et al., 2014; Shasha et al., 2015; Cate et al., 2013). These studies are essentially based on techniques and methods to detect the similarity between the profile elements. In this paper, the central focus is more on methods that are based on semantic similarity measures. In literature, there are methods based on simple or compound similarity measures (Zghal et al., 2015; Martínez-Villaseñor et al., 2014) some of which use the very costly supervised learning techniques (Shasha et al., 2015; Cate et al., 2013).

Hence, we propose a matching process resting on a classification process to (i) automatically detect the similarity between the profile elements and (ii) generate the mapping rules. We used the semi-supervised technique which is based on the unlabelled data in the classification process to improve the result accuracy (Zghal et al., 2013; Fazakis et al., 2016). Besides, this technique offers a predictive model adapted to the data to be classified with reduced labelled data. In addition to taking into account the semantic heterogeneity of the learner's profiles, our matching process aims, equally, at resolving the structural and the syntactic heterogeneity.

The rest of this paper is organized as follows. In Section 2, a state of the art about the learners' profiles interoperability system is introduced. Then, in Section 3, an overview about our contribution to the matter is presented. The proposed semi-supervised algorithm used to the matching process is identified in Section 4. The matching process is developed in Section 5. In Section 6, the evaluation results are displayed. Section 7 stands for the concluding part of the paper that sets forward an overview of the ongoing research that we are trying to accomplish.

2 State of the art

In this section, we present a brief overview of the users' profiles interoperability approaches. We also depict the matching process steps and techniques used by the majority of these systems. In addition, we describe the similarity measurement types and the methods used in the matching process followed by a synthesis.

2.1 *The learner's profiles interoperability approaches*

In order to implement a learner's profiles interoperability system, three main keys should be considered: the interoperability architecture, the data exchange representation and the data integration (Ghorbel et al., 2015). In this paper, we are interested in the second key which centers around how the exchanged data are represented.

Generally, the exchanged data representation is governed by the definition of standard ontology or unified profiles that can be used by multiple systems (Brusilovsky and Millan, 2007; Musa and Oliveira, 2005; Wagner et al., 2014; Ouf et al., 2017). This representation is an incomplete solution because of the variability of the stored learner's profile data (interests, preferences, historic navigation, evaluation ...) and the large number of differences in syntax, structure and semantics of the learner's profiles. In fact, the

emergence of a new system requires the reconstruction of a new anthology or a new unified profile. For this reason, a second representation seems to be a possible solution to these problems.

The solution lies in the use of matching techniques to generate mapping rules (Conde et al., 2014; Wu et al., 2015). Therefore, to exchange data, each system must do the mapping with the profile of the other systems. The merit of this solution is that each system may adopt its own representation of the learners' profiles in terms of language and structure. Yet, each pair of systems needs to create mapping in both directions. Therefore, multiple mapping rules should be implemented for different learner's profile models. In addition, during the introduction of a new learner's profile model representation, new mapping rules need to be developed.

The third representation seems to overcome the drawbacks of the first two ones. For this reason, we are rather interested in this representation which is essentially based on a common learner's profile and a matching process (Walsh et al., 2013; Martínez-Villaseñor et al., 2014; Santos and Boticario, 2015; Martínez-Villaseñor and González-Mendoza, 2017).

The profile matching is a schema manipulating process which takes as input two heterogeneous schemas and returns the mapping rules (Sutanta et al., 2016). Each mapping rule identifies the semantic relation between two schema elements. This identification relies on matching techniques and similarity measures.

The matching process takes place in three steps (Sutanta et al., 2016). The first one is for schema information extraction (element, name, hierarchical relations, structural and semantic relations). The second step consists in calculating the similarity between two schema elements using similarity functions. In the third step, the mapping rules are detected and validated on the basis of the calculated similarities. Roughly speaking, the validation is manually (Walsh et al., 2013; Sah and Wade, 2016) or semi-automatically performed (Jaiswal et al., 2013) which leads to a waste of time. For this reason, several systems, with an automatic validation, are suggested (Martínez-Villaseñor et al., 2014; Zghal et al., 2015; Martínez-Villaseñor and González-Mendoza, 2017).

These systems are based on linguistic and structural validation techniques. The linguistic technique is based on a syntactic or semantic similarity comparison. This technique takes into account just the meaning of the element name (word). However, the relations between the elements, such as neighbours, parents and descendants, are not taken into account. For this reason, the structural technique has emerged. It considers that the similarity between two elements e_1 and e_2 belonging respectively to schema S_1 and S_2 depends on the relationship between elements having a relation with e_1 and e_2 . For example, the authors in Martínez-Villaseñor et al. (2014) and Martínez-Villaseñor and González-Mendoza (2017) state that two elements are similar only if their neighbours are similar. However, the authors in Zghal et al. (2015) think that two elements are similar only if their parents or descendants are similar.

2.2 *The similarity comparison*

In order to compare the similarity of two words, sentences or documents, several types and methods are distinguished. In literature, the following similarity types are recorded: the string and the semantic similarity (Gomaa and Fahmy, 2013). In the string-based similarity type, we identify the character and the term measure. In the semantic-based similarity type, we identify the knowledge and the corpus similarity types.

The authors in Goma and Fahmy (2013) assume that knowledge-based similarity measures: (i) give an efficient semantic result compared to the string-based ones and (ii) provide a gain of time by giving the measure result compared to the corpus-based ones. Hence, we are interested in this type that can be divided into two groups of measures to identify the similarity degree between words: semantic similarity and relatedness. In the first group, there are three measures: Hirst and StOnge (hso), Lesk (lesk) and vector (vector). In the second group, we identify two sub-groups which are based on the information content and the path length measures. There are three measures of information content: Resnik (res), Lin (lin), Jiang and Conrath (jcn) and three path length measures: Leacock & Chodorow (lch), Wu and Palmer (wup) and Path Length (path).

Based on the identified types, several methods of similarity measurement have emerged. We classify them into three main methods.

The first one consists in using one of the measures relative to each type. Several researchers, such as Zghal et al. (2015), Mezghani et al. (2014), adopted this method. These researchers have used a similarity measure based on WorldNet in order to respectively identify the similarity between the document elements (Zghal et al., 2015) and the users' closed friend tags and their own tags (Mezghani et al., 2014).

The second method relies on a combination of two or more measures (Martínez-Villaseñor et al., 2014; Wang et al., 2014; Martínez-Villaseñor and González-Mendoza, 2017). The authors in Martínez-Villaseñor et al. (2014) and Martínez-Villaseñor and González-Mendoza (2017) used a combination of three similarity measures: the two first are character-based and the third is knowledge-based similarity measures. The authors in Wang et al. (2014) proposed a new similarity function called 'fuzzy-token-matching-based-similarity' which extends the string similarity functions (jaccard, cosine, etc.) by allowing fuzzy matching between two tokens. Then, they joined the new function in order to find the similar string pair.

The third method rests on the machine learning algorithm. Most of the studies using this method are based on the supervised learning techniques (Buscald et al., 2013; Severyn et al., 2013; Shasha et al., 2015) so as to classify text pairs. The researchers in Buscald et al. (2013) and Severyn et al. (2013) used the Support Vector Regression algorithm SVR and took several simple similarity measures as features for the learning process. Shasha et al. (2015) applied the multinomial naïve Bayes tree algorithm on a large number of widely used text classification benchmark datasets.

2.3 *Synthesis*

Departing from this state of the art, we have attempted to perform a comparative study of the learner's profile interoperability approaches (Table 1). The comparison rests on the mode of representation of the exchanged data, the matching techniques and the similarity types and methods used in the matching process. We noticed that most of the proposed approaches used the structural and/or the linguistic (Conde et al., 2014; Santos and Boticario, 2015; Martínez-Villaseñor et al., 2014) matching techniques in order to improve the result of the mapping rules. Moreover, it can be noticed that different types and methods of similarity comparison are used in the matching process. However, to the best of our knowledge, the majority of the approaches did not use the machine learning algorithm despite its importance in similarity comparison such as in textual similarity (Shasha et al., 2015; Cate et al., 2013). The majority of these approaches used the supervised machine learning algorithms which are costly in time (consuming), storage capacity (large quantity of labelled data), etc.

Table 1 Learner's profile interoperability approach comparative study

Approaches	Representation	Matching									
		Types		Similarity		Methods		Techniques			
		Character	String	Corpus	Simple	Combination	Machine Learning	Manual	Structural Linguistic		
Musa and Oliveira (2005); Wagner et al. (2014); Ouf et al. (2017)	Ontology standard	-	-	-	-	-	-	-	-	-	-
Conde et al. (2014); Wu et al. (2015)	Matching between profiles	+	-	-	+	-	-	-	-	-	+
Martínez-Villaseñor et al. (2014); Martínez-Villaseñor and González-Mendoza (2017)	Ontology standard and matching	-	+	-	-	+	-	-	-	-	+
Walsh et al. (2013); Sah and Wade (2016)	Unified profile and matching	-	-	-	-	-	-	-	+	-	-
Santos and Boticario (2015)		+	-	-	+	-	-	-	-	-	+

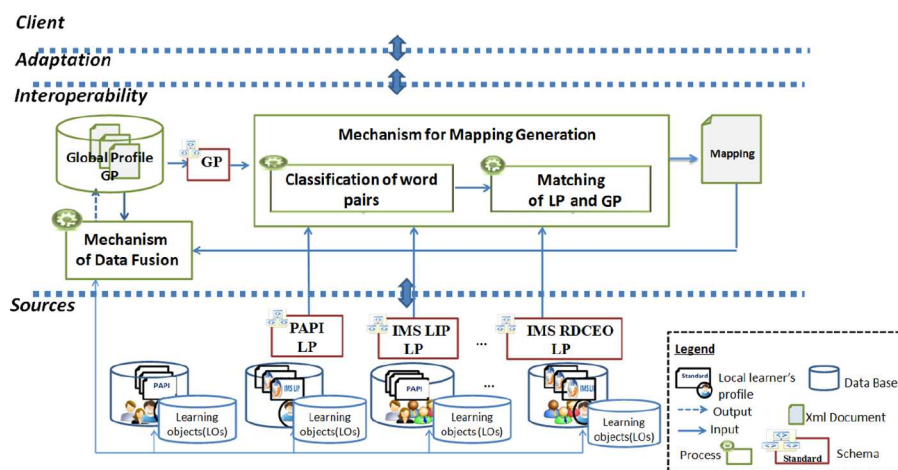
As a matter of fact, further efforts need to be made in order to improve the matching process and the mapping rules.

For this reason, we propose a mechanism for mapping generation based on a process for the learner's profiles matching using the semi-supervised machine learning technique. By using this technique, our mechanism is adapted to all provided heterogeneous learner's profiles.

3 Overview of the proposed mechanism for mapping generation

In order to allow the exchange of the heterogeneous learner's profile data between e-learning systems, we set forward an architecture for the learner's profiles interoperability. This architecture is illustrated in Figure 1. It is made up of four layers inspired from Ghorbel et al. (2015): client, adaptation, interoperability and sources.

Figure 1 The extended interoperability layer of our architecture (see online version for colours)



In our context, the client layer allows the interaction between the learner and e-learning systems. Thus, the learner can send a request by clicking on the provided links through different types of devices (PC, mobile, ...).

In the source layer, each system has its own local databases of Learning Objects (LOs) and Learners' Profiles (LPs). LOs can be web pages, images, videos, tests or any other elements involved in the learning process. They can be represented by different standards such as Learning Object Metadata³ (LOM), Information Message Service Simple Sequencing⁴ (IMS SS) and IMS Learning Design.⁵

Similarly, LPs can be represented by different standards such as PAPI, IMS LIP and IMS RDCEO. The learner can have a Local Profile (LP) in each educational system. In order to return the adapted results (LOs) to the learner based on the adaptation layer (Ghorbel et al., 2015), we need to exchange the data between his/her Local Profiles. Thus, we need to interoperate the heterogeneous learner's profile standards in the interoperability layer. In this layer, we are based on the third mode of representation of the exchanged data (Section 2.1): each LP refers to a common profile called Global Profile (GP) with a mechanism for mapping generation. GP helps the learner when navigating in LOs (lesson, activity ...) by taking into account several parameters. These parameters are detailed in

Section 5.1. The mechanism for mapping generation generates the mapping rules which serve to merge the exchanged data from LP to GP and vice versa based on a mechanism of data fusion (Ghorbel et al., 2016). This mechanism allows mainly the resolution of conflicts which may occur in the exchanged data .

In this paper, the extent of the interoperability layer is detailed using the mechanism for mapping generation. This mechanism, as shown in Figure 1, takes as input the schema of the Global Profile (GP) and the Local Profiles (LPs). Each LP is represented by a standard (e.g. PAPI, IMS LIP, etc.). It is governed by two processes in order to generate the mapping rules as an output. These processes are called matching of LP and GP and classification of word pairs.

The matching process uses the structural and the linguistic validation techniques (Section 2.1). For the structural technique, we take into account the hierarchical information. Therefore, two nodes are considered to be similar only if their parents are similar (Zghal et al., 2015). However, with the linguistic techniques, the matching process can detect the semantic similarity between two node names, which we have called word pair. The first word in the pair represents an LP node name and the second word represents GP node name. Each word pair is classified into similar or non-similar according to the classification of word pair process. This process is based on the semi-supervised learning technique.

In literature, several algorithms for semi-supervised learning are proposed (Fazakis et al., 2016), such as Self-Training, Co-Training, Tri-Training, semi-supervised Support Vector Machine (S^3VM) and Transductive Support Vector Machine (T-SVM). Some authors (Xiaojin, 2008; Fazakis et al., 2016) assert that Tri-Training: (i) is simpler than the last two mentioned algorithms and (ii) improves the classification result against all algorithms.

The classic Tri-Training algorithm (Xiaojin, 2008) trains three classifiers on the same labelled learning database based on its all features. The elements of the labelled learning database are selected randomly for training. Then, the elements of a nonlabelled database are classified. If two classifiers agree on the label of an element 'x' from the nonlabelled database, 'x' should be added to the learning database as a confident element to train the third classifier. This process is iterated until the three classifiers match the labelling of each element of the nonlabelled database.

This algorithm is sensitive to the random element selection and the use of the whole set of features for training. We regard that the projection of the database feature set into conditionally independent sub-sets of features (i.e. each sub-set of features is sufficient to train a classifier) for training as a key to resolve the notion of sensitivity. This concept of feature projection is inspired from Co-Training algorithm. This algorithm has shown accurate results (Zghal et al., 2013). In this paper, we propose a new Tri-Training algorithm version in order to improve the classification result accuracy.

4 Classification of word pairs

This process allows the identification of the similarity between LP and GP node names (word pair) with a new version of the semi-supervised Tri-Training algorithm.

4.1 Principle of the new Tri-Training algorithm

This algorithm is based on two learning databases taking the form of a matrix.

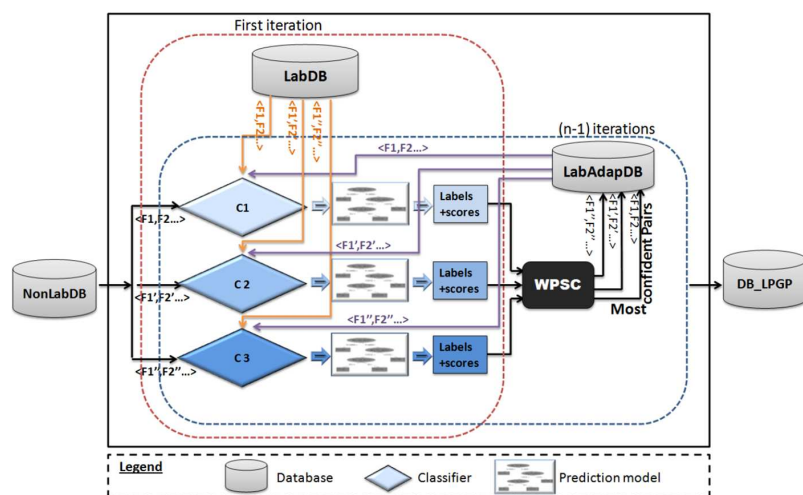
The first one is called Labeled Database (LabDB). Each line of LabDB contains: (i) a pair of similar or nonsimilar word pairs selected randomly, (ii) the semantic

knowledge-based similarity measure metrics (Section 2.2) which are used as features in the classification process and (iii) a label: the label ‘+’ if the pair is similar and the label ‘-’ if it is not.

The second learning database is called Non-Labeled Database (NonLabDB) having the same structure of LabDB. However, it is not labelled and in each line, we find : (i) a word pair relative to LP and GP node names and (ii) the semantic knowledge-based similarity measure metrics. Each word pair in NonLabDB is classified into similar or non similar with the new Tri-Training algorithm resting on LabDB.

Our Tri-Training algorithm consists in training three classifiers during nbIteration on: (i) LabDB or the labelled and adapted learning database (LabAdapDB), (ii) NonLabDB based on three independent feature sets ($\langle F^1, F^2, \dots \rangle$, $\langle F'^1, F'^2, \dots \rangle$, $\langle F''^1, F''^2, \dots \rangle$: each classifier takes a feature set (Figure 2) .

Figure 2 Word pair classification process based on the new Tri-Training algorithm (see online version for colours)



In fact, the features which are related to the knowledge-based similarity measures, can be divided into three sets (Section 2.2). The first set is related to the information content measures (res, lin and jcn). The second set is related to the path length measures (lch, wup and path). The third set is related to the semantic relatedness measures (hso and lesk).

In the first iteration, each classifier is trained on LabDB based on its relative feature set. Then, three prediction models are generated through which the classifiers assign the labels and scores to NonLabDB word pairs. Consequently, three results for each word pair are obtained. These results are combined in order to obtain one label for each word pair. Then, the most confident word pairs are detected and added to LabAdapDB.

During the following n-1 iterations, the classifiers are trained on LabAdapDB and the most confident word pairs are added to LabAdapDB. In order to adapt the latter, we propose an algorithm called Word Pair Score Calculation (WPS).

4.2 Word pair score calculation

The WPS algorithm calculates the scores of each pair after each iteration: a score for having the label ‘+’ and another for having the label ‘-’. Then, it detects the most confident word pairs (see Algorithm 1) and add them to LabAdapDB.

Algorithm 1 *Word Pair Score Calculation: WPSC***Input:**

nbIteration, nbClassifier : Integer,
LabDB, LabAdapDB, NonLabDB : Database,
feature : Array of size(*nbClassifier*),
C : Object of a Supervised classifier type ,
sumScore, score, finalResult : matrices of size(
sizeOf(NonLabDB)X2)

Output: *DB_SDGP* : Database

```
1: Initialize Matrix sumScore[0..sizeOf(NonLabDB), 0..2] into zeros;  
2: for i ← 1 to nbIteration do  
3:   if (i = 1) then  
4:     base ← LabAdapDB;  
5:   else  
6:     base ← LabDB;  
7:   end if  
8:   for cl ← 1 to nbClassifier do  
9:     score ← classify(C, NonLabDB, base, feature[cl]);  
10:    for k ← 1 to sizeOf(NonLabDB) do  
11:      add(sumScore[k, 1], score[k, 1]);  
12:      add(sumScore[k, 2], score[k, 2]);  
13:      if (cl == nbClassifier) then  
14:        if (sumScore[k, 1] >= sumScore[k, 2]) then  
15:          add(finalResult[k, 1], sumScore[k, 1]);  
16:          add(finalResult[k, 2], '+');  
17:        else  
18:          add(finalResult[k, 1], sumScore[k, 2]);  
19:          add(finalResult[k, 2], '-');  
20:        end if  
21:      end if  
22:    end for  
23:  end for  
24:  listMax_Index ← research_MaxIndex(finalResult);  
25:  addConfidentPairs(listMax_Index, NonLabDB, finalResult,  
LabAdapDB);  
26: end for  
27: DB_LPGP ← addLabel(finalResult, NonLabDB);
```

end

The inputs of this algorithm are the number of iterations (*nbIteration*), the number of classifiers (*nbClassifier*), *LabDB* and *LabAdapDB*, the array of the sub-sets of features (*feature*), the supervised classifier (*C*), the matrices which store scores and labels (*sumScore*, *score* and *finalResult*).

In each iteration (line 1), a number of results relative to *nbClassifier* (three in our case) are obtained based on *LabDB* or *LabAdapDB* (lines 1–9) and their relative sub-sets of features.

By using a sub-set of features (*feature*[*cl*]), *C* assigns to each word pair two scores for each label based on the function *classify* (line 9). The result of each classifier is stored in

the matrix score. This matrix contains in columns 1 and 2 the scores of each word pair so as to acquire respectively the label '+' and the label '-'.

Then, these scores are added to the matrix `sumScore` (lines 11–12) in order to compute the sum of the scores for each word pair so as to acquire the label '+' (`sumScore[k,1]`) and the label '-' (`sumScore[k,2]`). Therefore, `sumScore` is used in order to identify the final score and the label for each word pair by comparing the assigned scores (line 14) in the first and the second column (`sumScore[k,1]` and `sumScore[k,2]`). Therefore the final scores and labels are stored in the matrix `finalResult` (lines 13–19).

The latter is used in order to adapt `LabAdapDB`. Afterwards, the algorithm detects the most confident word pairs in `NonLabDB` by using the function `research_MaxIndex` (line 24). This function detects the indexes (`listMax_Index`) of pairs having the maximum score in `finalResult`. Then, based on the detected list, the algorithm adds these confident word pairs to `LabAdapDB` with its labels based on the function `addConfidentPairs` (line 25). Finally, a database `DB_LPGP` containing the word pairs of `NonLabDB` and its final labels is generated (line 27). `DB_LPGP` is used after in the matching process.

5 Matching of local and global profiles

The learner's profiles can respect different syntaxes such as Resource Description Framework (RDF), Extensible Markup Language (XML) and Web Ontology Language (OWL).

For this reason, before starting the matching process, we perform a preliminary step, which is the profile schema information extraction. Then, we propose an algorithm for matching of the Global Profile (GP) and a Local Profile (LP).

5.1 Schema information extraction

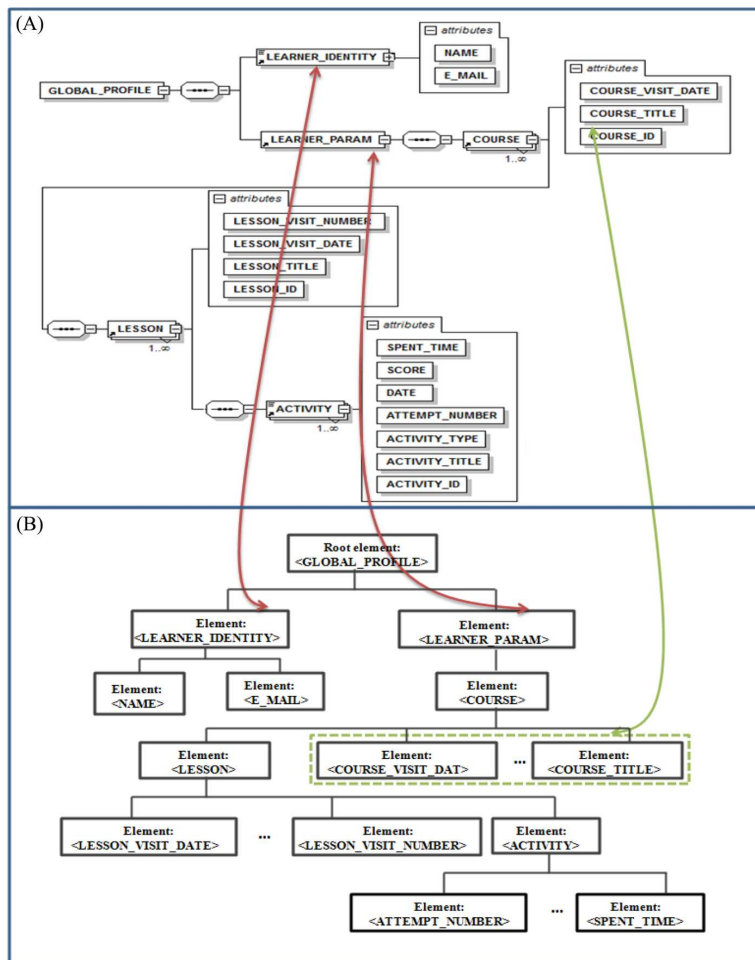
Some works handle this step in order to resolve the syntactic heterogeneity of schema document description by the extraction of linguistic and structural information. Similarly, in our work we attempt to extract: (i) the linguistic information which stands for the name of elements, attributes and properties. and (ii) the structural information identified by its hierarchy. Then, based on the extracted information, a tree for each schema is constructed based on the Document Object Model⁶ (DOM) standard. In this standard, every element, attribute/property is respectively an element, attribute/property node with its corresponding text (name). However, in our work, attributes and properties are represented as element nodes with their names. All these nodes are contained within the root node of the tree.

Figure 3 represents GP XML schema (Figure 3A) and its corresponding tree (Figure 3B). GP is proposed in the interoperability layer of our architecture (Section 3). It helps the learner when navigating in Learning Objects (LOs) by taking into account several parameters.

Departing from Figure 3A, the learner has two main parameters: the first portrays his/her identity (`LEARNER_IDENTITY`) and the second describes his/her learning experience (`LEARNER_PARAM`). The learning experience is made up of the visited courses (`COURSE`). Each (`COURSE`) is identified by (`COURSE_ID`), its name (`COURSE_TITLE`), the last visit date (`COURSE_VISIT_DATE`). For each (`COURSE`), we specify the learned lessons and their related activities. A (`LESSON`) is identified by (`LESSON_ID`), its name (`LESSON_TITLE`), the last visit date (`LESSON_VISIT_DATE`) and the number of visits (`LESSON_VISIT_NUMBERB`). An activity is identified by (`ACTIVITY_ID`), its name (`ACTIVITY_TITLE`), its type (`ACTIVITY_TYPE`), the time spent (`SPENT_TIME`), the date

of the visit (DATE), the number of attempts (ATTEMPT_NUMBER) and the attained score (SCORE).

Figure 3 The Global Profile XML Schema and its corresponding tree (see online version for colours)



All these elements and attributes are transformed into a tree in Figure 3B. Each element and its corresponding attributes are represented as a node element. The root node, named *<Global_Profile>*, holds two nodes: *<LEARNER_IDENTITY>* and *<LEARNER_PARAM>*. The second node holds the child node *<COURSE>* which holds the child nodes *<COURSE_VISIT_DATE>*, *<LESSON>*, etc.

5.2 The matching algorithm

After the information extraction step, we proceed with matching the tree relative to LP and GP based on the matching of trees algorithm (see Algorithm 2). The result of this algorithm is a set of mapping rules between LP and GP.

Algorithm 2 Matching of trees

Input: $LP, GP : tree, DB_LPGP : Database$

Output: $mapping : XML File$

```
1:  $list\_ndLP = root(LP).getChildren();$ 
2:  $list\_ndGP = root(GP).getChildren();$ 
3: for each  $node\ ndLP$  of  $list\_ndLP$  do
4:   for each  $node\ ndGP$  of  $list\_ndGP$  do
5:      $match(ndLP, ndGP);$ 
6:   end for
7: end for
8: Procedure  $match(ndLP, ndGP)$ 
9: Begin
10:  $list\_ndLP = ndLP.getChildren();$ 
11:  $list\_ndGP = ndGP.getChildren();$ 
12: if ( $nodeName(ndLP).equals(nodeName(ndGP))$ )
    ||  $similarity(nodeName(ndLP), nodeName(ndGP), DB\_LPGP)$ ) then
13:    $add(nodeName(ndLP), nodeName(ndGP), mapping);$ 
14:   if ( $list\_ndLP \neq null \ \& \ list\_ndGP \neq null$ ) then;
15:     for each  $node\ nl$  of  $list\_ndLP$  do
16:       for each  $node\ ng$  of  $list\_ndGP$  do
17:          $match(nl, ng);$ 
18:       end for
19:     end for
20:   end if
21: else
22:   if ( $list\_ndGP \neq null$ ) then
23:     for each  $node\ ng$  of  $list\_ndGP$  do
24:        $match(ndLP, ng);$ 
25:     end for
26:   end if
27:   if ( $list\_ndLP \neq null$ ) then
28:     for each  $node\ nl$  of  $list\_ndLP$  do
29:        $match(nl, ndGP);$ 
30:     end for
31:   end if
32: end if
33: End match
34: Function  $similarity(ndLP, ndGP, DB\_LPGP)$ 
35: Begin
36:  $test = false;$ 
37: for each  $line$  in  $DB\_LPGP$  do
38:   if ( $line.contains(nodeName(ndLP), nodeName(ndGP), '+' )$ ) then
39:      $test = true;$ 
40:   end if
41: end for
42:  $return\ test;$ 
43: end similarity
```

end

This algorithm takes as input the tree of LP and GP and a database DB_LP GP. First, this algorithm performs the matching between the children of the root of LP (line 1) and GP (line 2) with the call of the recursive procedure match (lines 1–7). This procedure performs the matching of the other nodes.

It takes two parameters: the first and the second are respectively a node of LP (ndLP) and GP (ndGP). Then, match verifies the similarity between the node names of LP and GP (line 7) based on exact comparison or function similarity (line 8). This function takes three parameters: the nodes ndLP and ndGP and the database DB_LP GP. In fact, the originality of our matching process resides in using this database which is the result of our new Tri-Training algorithm (Section 4). This database contains all the possible combinations between the LP and the GP node names (pairs of words). Each line in DB_LP GP represents a pair of node names and its label: the label ‘+’ for a similar pair of node names and the label ‘-’ for non-similar one. For this reason, the function similarity checks if two introduced node names exist in DB_LP GP and are similar or not (lines 34–43).

In case of similarity, the node names are added as new nodes in an XML file called mapping (line 13). Then, if ndLP and ndGP has children nodes (line 14), the match procedure restarts by matching their children (lines 15–19).

In case of non-similarity and if ndGP has children, match restarts by matching ndLP with ndGP children (lines 22–26). In case of non-similarity and if ndLP has children, match restarts by matching ndLP children and ndGP (lines 27–32). At the end of this algorithm, a set of mapping rules are identified and stored in the XML file (mapping).

Figure 4, illustrates an extract of the obtained XML mapping file which contains: (i) an extract of the tree of the PAPI standard (LP = PAPI), IMS RDCEO standard (LP = RDCEO) and GP, (ii) the existing relationship between them and (iii) the mapping file. For example, for the node ‘RDCEO:period’, we save its synonym in GP in the node *< SimilarTo >* (*< SimilarTo name = GP : date >*). These two node names are similar because on the one hand they exist in DB_LP GP with the label ‘+’ and on the other hand, their parent nodes (RDCEO:activity and GP:ACTIVITY) are similar.

6 Evaluations

In this section, the datasets and the metrics used for the evaluation are described. Then, the obtained results are displayed.

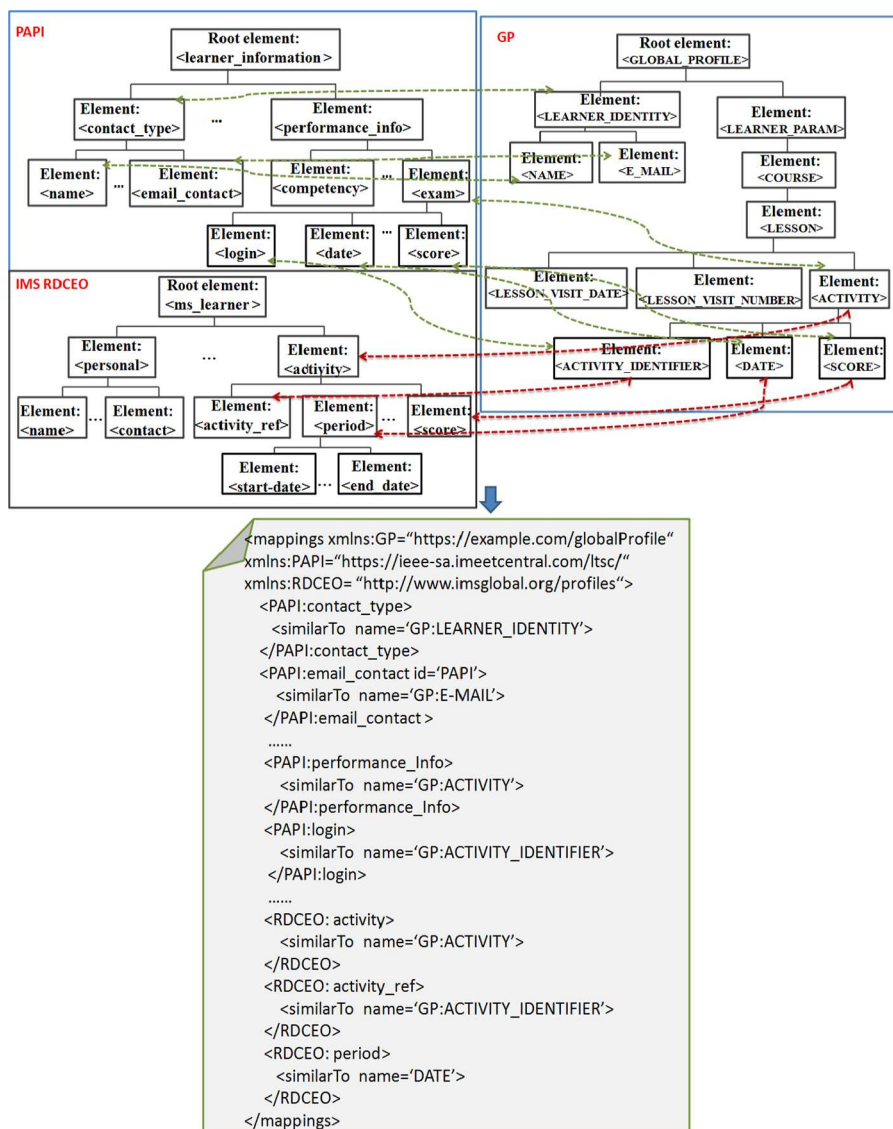
6.1 Datasets

In this evaluation, three distributed e-learning systems are selected: (i) two learning management systems called Moodle⁷ and Claroline⁸ and (ii) a learning assessment system called Position Platform.⁹ Learners are members of the three systems at the same time. In Moodle and Claroline, learners can learn courses, carry out activities, receive marks about these activities, sit for exams, etc. In the Position Platform, learners can sit for exams in the form of multiple choice questions and get marks.

In this paper, we studied the case of students of the Virtual University of Tunisia who desire to get a Certificate in Information technologies and Internet (C2I). The courses of

this module belong to 5 domains and each domain includes 4 competencies (sub-domains) each of which includes several themes. Once a student desires to get the certificate exam, he/she accesses his/her account on Moodle for revision where he finds several links related to the whole domains (competencies, activities, etc.). Some of these links are not useful. Each student needs to be oriented with the best links relative to different learning objects (LOs) in order to accomplish his/her revision based on data stored in his/her three profiles (example: links to domains or sub_domains where the student does not have good marks in the related activities and the related passed exams).

Figure 4 Example of PAPI and GP matching and the generated mapping rules (see online version for colours)



The data exchange (marks, spent time in activities, number of attempts, etc.) of each student in such domain (competency) in Claroline and the Position Platform with Moodle deals with certain difficulties because the learner's profiles schemas are different.

In fact, the learner's profile in Moodle, Claroline and the Position Platform are represented respectively by the IMS LIP, the IMS RDCEO and the PAPI standard (Section 1).

The IMS LIP standard includes several structured categories: identification; goal; qcl (qualifications, certifications, licenses); activity; competency; transcript; accessibility; interest; affiliation; security keys; and relationship. The identification category represents the demographic and biographic data about a learner. The goal category represents the learning career and other objectives of a learner. The qcl category is used to identify the qualifications, certifications, and licenses from recognized authorities. The activity category involves the learning related activities in any state of completion. The competency category stands for skills, experience and acquired knowledge. The transcript category represents the institutional academic achievements. The accessibility category introduces the general accessibility to the learner information by means of language capabilities, disabilities, eligibility, and learning preferences. The interest category describes hobbies, recreational activities, etc. The affiliation category represents information records about membership in professional organizations. The security key sets the passwords and keys assigned to a learner. The relationship category establishes the link between core data elements.

Moreover, the IMS RDCEO standard involves improved elements of existing categories in the IMS LIP standard and other new categories namely profile, educational path, metadata, and comment (Oubahssi and Grandbastien, 2006). The profile category includes a set of information gathered as an output of a learning unit. The metadata category assembles the data, which makes it possible to describe a learner's data. The educational path category displays the educational steps carried out by a learner during his/her training. The comment category is used for adding information or remarks to complete the category elements.

The PAPI standard provides a classification of the learner's information according to 6 categories. The first category, which is called `contact_type`, describes personal information, such as the name, `e_mail_contact`, `postal_adress`, etc. The second one displays the relationship information (`relations_info`), such as the learner's relations with other learners or tutors. The third exhibits the security information (`security_info`). The fourth and the fifth portrays the learner's preferences (`preference_info`) and performances (`performance_info`). The latter includes the learner's goals, experience and the work in progress. The last category (`portfolio_info`) presents the learner's competencies and a collection of the performed tasks.

6.2 *Evaluation metrics*

To address the heterogeneity of the mentioned standards, our architecture was set up between the three systems. Then, a set of evaluations of our proposed mechanism for mapping generation is performed. First, the classification of word pair process is evaluated. For this reason, the classic metrics for classification evaluation (Powers, 2011) which are Classification Rate (CR) and F-measure (FR) Rate are used.

Second, the matching of Global Profile (GP) and Local Profile (LP) process is evaluated. At a first stage, the response time of matching is assessed. Then, the generated mapping rules are evaluated based on Precision, Recall and F-measure.

Precision is the proportion of correct mapping rules among those returned by the mechanism.

$$Precision = \frac{|detectedMapping \cap realMapping|}{|detectedMapping|} \quad (1)$$

Recall is the proportion of correct mapping returned by the mechanism among the correct ones (including the correct mapping rules that are not detected by the mechanism). This value indicates the effectiveness of the mechanism and demonstrates the percentage of the missed mapping rules

$$Recall = \frac{|detectedMapping \cap realMapping|}{|realMapping|} \quad (2)$$

F-measure demonstrates the quality of our mechanism by combining the precision and the recall.

$$F-measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

We carried out these evaluation on a PC having 4G of RAM and 1.85 GHz processor.

6.3 Evaluation of the classification process

In this evaluation, we performed a comparison of the obtained classification results based on the Co-Training, the classic and the new Tri-Training algorithm and the manual labelling. For this reason, we have taken into account three NonLabDB and a Labeled Database (LabDB) (Section 4.1). The first NonLabDB contains 500 nonlabelled pairs of words from the Global Profile GP and the PAPI standard called *NonLabDB_{PAPI}*. The second NonLabDB contains 500 pairs of words from the GP and IMS LIP standard called *NonLabDB_{IMSLIP}*. The third NonLabDB contains 900 pairs of words from the GP and IMS RDCEO standard called *NonLabDB_{RDCEO}*. These three databases are labelled manually for evaluation. LabDB contains 50 labelled pairs of words.

After data preparation, we proceeded with the choice of the best classification algorithm (classifier C) for training. For this reason, we applied the Co-Training, the classic Tri-Training and the new Tri-Training version using several supervised classification algorithms, such as RandomTree and Support Vector Machine.

Figure 5 illustrates the new Tri-Training algorithm average evaluation results by applying the RepTree, RandomTree, Support Vector Machine and Bayesian networks for the three NonLabDB. We noticed that the RandomTree algorithm is the best classifier with the highest CR and FR values (0.85 and 0.84).

In order to identify the result of the Co-Training algorithm, we performed three evaluations based on three projections of the three NonLabDB features. The first projection is based on the Information Content measures and Path measures. The second rests on the Information Content and Relatedness measures whereas the third is based on the Path and Relatedness measures.

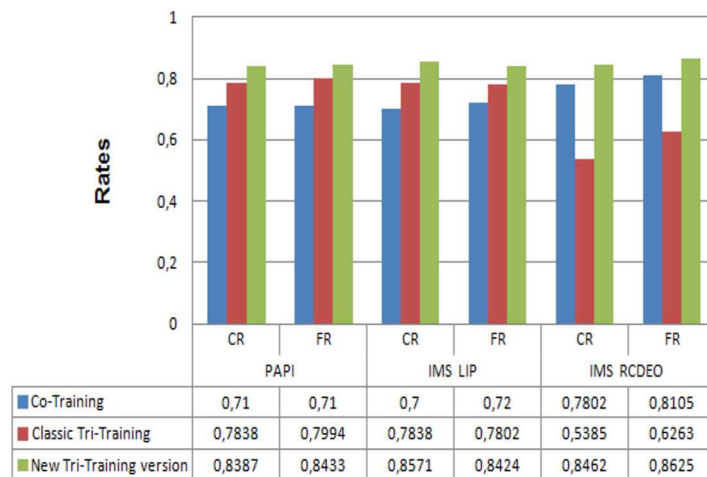
Figure 5 The Tri-Training evaluation result using several classification algorithms (see online version for colours)



Moreover, we evaluated the classic Tri-Training algorithm for the three NonLabDB. The results highlight that the values of the CR and FR are higher than the ones obtained with the Co-Training. In addition, these results are further improved with the application of our proposed Tri-Training algorithm.

Figure 6 presents the results of the comparison between the Co-Training, the classic Tri-Training and our proposed Tri-Training algorithm.

Figure 6 Comparative results of Co-Training, classic Tri-Training and new Tri-Training version (see online version for colours)



This figure displays a clear result improvement. In fact, the average of CR and FR for the three NonLabDB increased from 0.73, 0.74 (with the Co-Training) and 0.70, 0.73 (with the classic Tri-Training) to 0.85, 0.84 (with the new Tri-Training version). Therefore, the

results obtained in Figure 6 prove the efficiency of our proposed Tri-Training algorithm and can confirm the correctness and the validity of the majority mapping rules.

6.4 Evaluation of the matching process

This evaluation was conducted in order to prove the efficiency of the Matching of GP and LP resting on our new Tri-Training algorithm.

First, we assessed the response time of matching based on our new Tri-Training algorithm and the simple similarity measures. The response time includes: (i) the time for the information extraction of the schemas of LP and GP and (ii) comparing the similarity between node names based on the simple similarity measures such as wup and lin (Section 2.2) or the database DB_LPGP labelled by our new Tri-Training algorithm (Section 4.2) and (iii) creating the XML mapping file containing the mapping rules (Section 5.2).

The results are illustrated in Table 2 and show a clear gain in matching response time for generating the mapping rules for PAPI and GP, IMS LIP and GP in addition to IMS RDCEO and GP. For example, the generation of the total mapping rules between PAPI and GP takes a time of 2937 ms based on the simple similarity measure wup, which decreases to 1488 ms, based on our proposed mechanism for mapping generation with our Tri-Training algorithm. Therefore, we have a gain of 1449 ms. Resting on the obtained results in Table 2, we recorded a gain between 47 and 60% in the response time.

Table 2 Response time matching results

LP/GP	Response time in ms based on:	
	Simple Similarity Measure (wup, lin, lch...)	Our proposed Tri-Training algorithm
PAPI/GP	>2000 and <3000	1488
IMS LIP/GP	> 6000 and <7000	2410
IMS RDCEO/GP	>6000 and <7000	2500

Second, we assessed the generated mapping rules with the metrics defined in Section 6.2.

With these metrics, we tested the generated mapping rules coming from the PAPI and GP, IMS LIP and GP in addition to IMS RDCEO and GP. The result values are depicted in Table 3. These values confirm the medium-high quality of the generated mapping rules (0.8 for the F-measure average).

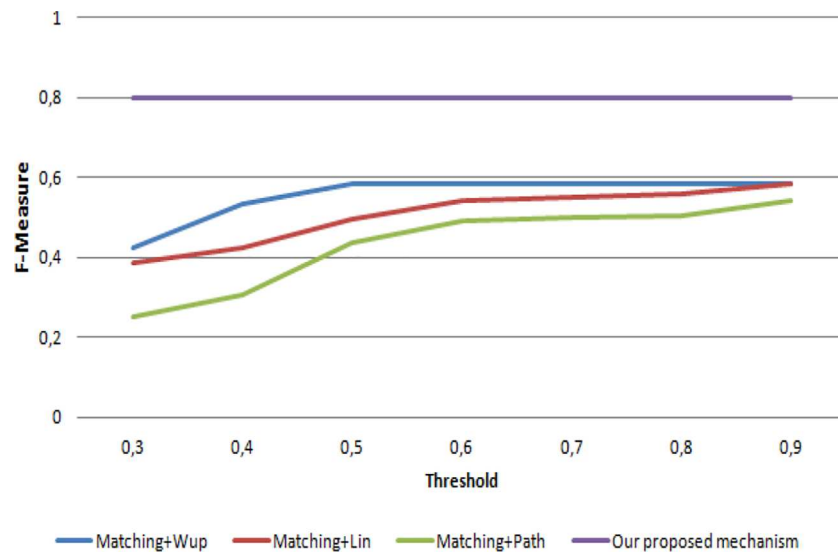
Table 3 Evaluation of the generated mapping rules between GP and LP

LP/GP	Precision	Recall	F-measure
PAPI/GP	0.64	1	0.78
IMS LIP/GP	0.70	1	0.82
IMS RDCEO/GP	1	0.68	0.80

Finally, we checked the performance of the matching process by using different simple similarity measures, such as lin and wup (without the classification of word pairs process)

used by the majority of the state of the art matching approaches. Then, we varied the threshold so as to determine whether two words are similar or not. We calculated the F-measure values (for each threshold) of the generated mapping rules (between GP and PAPI, GP and IMS LIP and GP and IMS RDCEO). Figure 7 reveals the average of results when using the simple similarity measure lin, wup and path with the matching process. We notice that the quality of the mapping rules is improved as the threshold increases. However, the generated mapping rules based on our mechanism remain the best ones with the highest F-Measure value (0.8).

Figure 7 Mapping rule quality comparative results (see online version for colours)



6.5 Evaluation on other datasets

This final evaluation was performed in order to prove that our mechanism for mapping generation can be applied for other purposes. For this reason, we took the case of the interoperability between the learner's profiles and the learning objects. In fact, during his/her learning experience, the learner can visit different learning objects (courses, lessons, activities, etc.). Some meta-data values about these learning objects need to be added to the Local Profile (LP) in order to enhance the result of adaptation. However, each Learning Object (LO) can be represented by a standard such as LOM, IMS SS, IMS LD (Section 3). For this reason our mechanism is set up between each LO and LP in order to generate a set of mapping rules. LP (IMS LIP, IMS RDCEO or PAPI) plays the role of the Global Profile GP (Figure 1) and LO plays the role of LP. The most popular standards for LO representation which are LOM and IMS SS are selected.

Table 4 portrays the evaluation results of the generated mapping rules for each LO and LP. These results confirm also (Figure 7) the medium-high quality of the generated mapping rules (0.72 for the F-measure average) and that our mechanism for mapping generation can be used for several purposes.

Table 4 Evaluation of the generated mapping rules between LO and LP

LO/LP	F-measure
LOM/IMS LIP	1
LOM/IMS RDCEO	0.8
LOM/PAPI	0.56
IMS SS/IMS LIP	0.61
IMS SS/IMS RDCEO	0.516
IMS SS/PAPI	0.88

7 Conclusion

In this paper, we set forward a mechanism for mapping generation based on the Tri-Training algorithm in order to match the learner's profiles represented by different standards, semantics and structures. This mechanism performs the matching between the Local Profiles (LPs) and the Global Profile (GP) by using the result of the classification of word pairs process. The latter is based on a new version of the semi-supervised Tri-Training algorithm. The efficient evaluation values denote that our mechanism not only improves the quality of the generated mapping rules but can also be used for several purposes.

In future research, we aspire to evaluate the effectiveness of our mechanism for mapping generation and our previously proposed mechanism of data fusion on the adaptation layer of our architecture.

References

- Brusilovsky, P. and Millan, E. (2007) 'User models for adaptive hypermedia and adaptive educational systems', *The Adaptive Web, Methods and Strategies of Web Personalization Lecture Notes in Computer Science*, pp.3–53.
- Buscald, D., Roux, J.L. and Flores, J.G. (2013) 'Lipn-core: semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features', *AI Magazine*, Vol. 17, No. 3, pp.37–54.
- Cate, B.T., Dalmau, V. and Kolaitis, P.G. (2013) 'Learning schema mappings', *ACM Transactions on Database Systems (TODS)*, Vol. 38, No. 4, p.28.
- Conde, M.Á., García-Peñalvo, F., Alier, M., Mayol, E. and Fernández-Llamas, C. (2014) 'Implementation and design of a service-based framework to integrate personal and institutional learning environments', *Science of Computer Programming*, Vol. 88, pp.41–53.
- Fazakis, N., Karlos, S., Kotsiantis, S. and Sgarbas, K. (2016) 'Self-trained lmt for semisupervised learning', *Computational Intelligence and Neuroscience*, Vol. 2016, p.10.
- Ghorbel, L., Zayani, A. and Amous, I. (2015) 'Improve the adaptation navigation in educational cross-systems', *KES: Knowledge-Based and Intelligent Information & Engineering Systems*, pp.662–670.
- Ghorbel, L., Zayani, C. A., Amous, I. and Sèdes, F. (2016). 'Learner's profile Hierarchization in an interoperable education system', In *International Conference on Intelligent Systems Design and Applications*, Part of the Advances in Intelligent Systems and Computing book series AISC, Springer, Vol. 557, pp.549–558.
- Gomaa, H.W. and Fahmy, A.A. (2013) 'A survey of text similarity approaches', *International Journal of Computer Applications*, Vol. 68, No. 13, pp.13–18.

- Jaiswal, A., Miller, D.J. and Mitra, P. (2013) 'Schema matching and embedded value mapping for databases with opaque column names and mixed continuous and discrete-valued data fields', *ACM Transactions on Database Systems (TODS)*, Vol. 38, No. 1, p.2.
- Martínez-Villaseñor, M.L. and González-Mendoza, M. (2017) 'Overview of a framework for ubiquitous user models interoperability', In *Applications for Future Internet*, Part of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering LNICTS, Springer, Vol. 179, pp.185–188.
- Martínez-Villaseñor, M.L., González-Mendoza, M. and Valle, I.D.D. (2014) 'Enrichment of learner profile with ubiquitous user model interoperability', *Computación y Sistemas*, Vol. 18, pp.359–374.
- Mezghani, M., Péninou, A., Zayani, A., Amous, I. and Sèdes, F. (2014) 'Analyzing tagged resources for social interests detection', in *16th International Conference on Enterprise Information Systems: ICEIS*, Lisbonne, Portugal, pp.340–345.
- Musa, D. and Oliveira, J. (2005) 'Sharing learner information through a web services-based learning architecture', *The Journal of Web Engineering*, Vol. 4, No. 3, pp.263–278.
- Oubahssi, L. and Grandbastien, M. (2006) 'From learner information packages to student models: Which continuum?', in *Intelligent Tutoring Systems*, Springer, Berlin/Heidelberg, pp.288–297.
- Ouf, S., Ellatif, M.A., Salama, S.E. and Helmy, Y. (2017) 'A proposed paradigm for smart learning environment based on semantic web', *Computers in Human Behavior*, Vol. 72, pp.796–818.
- Powers, D.M. (2011) 'Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation', *Journal of Machine Learning Technologies*, Vol. 2, No. 1, pp.37–63.
- Sah, M. and Wade, V. (2016) Personalized concept-based search on the linked open data. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 36, pp.32–57.
- Santos, O.C. and Boticario, J.G. (2015) 'Practical guidelines for designing and evaluating educationally oriented recommendations', *Computers & Education*, Vol. 81, pp.354–374.
- Severyn, A., Nicosia, M. and Moschitti, A. (2013) 'Learning semantic textual similarity with structural representations', in *51st Annual Meeting of the Association for Computational Linguistics*, 4–9 August, Sofia, Bulgaria, pp.714–718.
- Shasha, W., Liangxiao, J. and Chaoqun, L. (2015). 'Adapting naive bayes tree for text classification', *Knowledge and Information Systems*, Vol. 44, No. 1, pp.77–89.
- Sutanta, E., Wardoyo, R., Mustofa, K. and Winarko, E. (2016) 'A hybrid model schema matching using constraint-based and instance-based', *International Journal of Electrical and Computer Engineering*, Vol. 6, No. 3, p.1048.
- Wagner, A., Barbosa, J.L.V. and Barbosa, D.N.F. (2014) 'A model for profile management applied to ubiquitous learning environments', *Expert Systems with Applications*, Vol. 41, No. 4, pp.2023–2034.
- Walsh, E., O'Connor, A. and Wade, V. (2013) 'The fuse domain-aware approach to user model interoperability: A comparative study', in *IEEE 14th International Conference on Information Reuse and Integration (IRI)*, IEEE, pp.554–561.
- Wang, J., Li, G. and Feng, J. (2014) 'Extending string similarity join to tolerant fuzzy token matching', *ACM Transactions on Database Systems (TODS)*, Vol. 39, No. 1, p.7.
- Wu, D., Lu, J. and Zhang, G. (2015) 'A fuzzy tree matching-based personalized e-learning recommender system', *IEEE Transactions on Fuzzy Systems*, Vol. 23, No. 6, pp.2412–2426.
- Xiaojin, Z. (2008) 'Semi-supervised learning literature survey', *Computer Sciences Technical Report 1530 University of Wisconsin Madison*.
- Zghal, R., Ghorbel, L., Zayani, A. and Amous, I. (2013) 'An adaptive method for user profile learning', in *East European Conference on Advances in Databases and Information Systems*, Springer, Berlin/Heidelberg, pp.126–134.
- Zghal, R., Mnif, F., Zayani, C.A. and Amous, I. (2015) 'Adaptive global schema generation from heterogeneous metadata schemas', *Procedia Computer Science*, Vol. 60, pp.197–205.

Notes

- 1 <https://ieee-sa.imeetcentral.com/ltsc/>.
- 2 <http://www.imsglobal.org/profiles>.
- 3 <https://standards.ieee.org/findstds/standard/1484.12.1-2002.html>.
- 4 <https://www.imsglobal.org/simplesequencing/index.html>.
- 5 <https://www.imsglobal.org/learningdesign/index.html>.
- 6 http://www.w3schools.com/xml/xml_dom.asp
- 7 <https://moodle.org>.
- 8 www.claroline.net.
- 9 <http://c2i.uvt.rnu.tn/c2iv2>.