Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

Improving Feedbacks for ITS Assessment of Concept Maps

Hugo Traverson, David Genest, Stephane Loiseau
LERIA (Laboratoire d'Etude et de Recherche en Informatique d'Angers)
UFR Sciences, 2 Boulevard de Lavoisier, 49000 Angers FRANCE

Abstract

Assessment in intelligent tutoring system (ITS) on concept maps (CM) matches an expert CM to a learner CM. Feedbacks are provided to the learner as semantic comments and visual corrections. In this paper, quality of feedbacks is improved by using an ontological semantic for matching, formalized as a correlation feedback. Matchings are selected based on an overall assignment solution providing a suboptimal set of correlation feedbacks to the learner.

1 Introduction

Since the early 1970s, Intelligent Tutoring System (ITS) has been investigating computer-aided-instruction. In particular, the tutoring model is a branch centered on assessing the learner knowledge to provide appropriate feedbacks within a tutoring strategy. Assessment is based on comparing the expert knowledge as domain model with the learner's knowledge as student model. Within the range of ITS, this paper focuses on providing feedbacks on *Concept Map* (CM) (Novak and Gowin 1984), a representation language for both domain and student models. CM corresponds to directed multigraph composed of node set and arc set labeled respectively by CM *entities* as concepts and relations. CM knowledge is centered on *propositions* corresponding to the meaningful sentence of each arc and composed of a triple $<$ concept, relation, concept $>$.

CM assessment checks if a student's learning was meaningful by comparing expert and learner CMs and compute feedbacks for the learner relatively to the expert CM. Feedbacks can take the form of scoring (based on similarity measures (Strautmane 2012)), visual representations (e.g coloring) or textual comments. Feedbacks focus on two criteria: CM local content and overall structure. Concerning local content analysis, literature describes mostly similarity measures to match a pair of entities or propositions; for example (Harrison et al. 2004) uses syntactic synonyms based on WordNet, (Gouli et al. 2005) defines categories of errors (partially correct and superfluous or missing proposition/concept from the learner CM) and (Cline, Brewster, and Fell 2010) adds a manually created semantic to entities. Concerning overall structure analysis, works focus on studying CM as a whole and analysing structure patterns : (Novak and Gowin 1984) scores tree-like levels of hierarchy, (McClure, Sonak, and Suen 1999) computes holistic score, (Limongelli et al. 2016) defines a set of seven measures to compare CMs and (Soika and Reiska 2014) defines disciplinary topics. There are also papers describing systems or softwares mixing both criterias: (Schwendimann 2011) defines a multi-level system (KIS), (Marshall, Chen, and Madhusudan 2006) presents a similarity flooding algorithm on neighbor nodes to provide learner a set of feedbacks (Alves da Silva et al. 2012) uses ontology alignment as an assessment tool, (De Souza et al. 2008) uses graph isomorphism problem technique and (Gouli et al. 2005) computes quantitatively proposition assignments. On local content, as said in (Kharatmal and Nagarjuna 2006), ambiguity caused by CM free semantics limits range and quality of feedbacks provided by assessment. Most papers add external semantics to improve matchings of entities or propositions, whether it be by an ontology, a thesaurus like wordnet or any external semantic informations. However, they do not adapt external semantic measures to CM features. Using ontologies, either they transform CM in ontology and base assessment on alignment methods (Park and Calvo 2008; Alves da Silva et al. 2012; Graudina, Grundspenkis, and Milasevica 2012) or multiply expert CM representations to have different comparison views (Da Rocha, Favero, and Da Costa Junior 2008). On overall structure, as noted by (Canas, Novak, and Reiska 2015), an excellent CM describes a topic in a clear fashion, selecting especially a set of key topic propositions. Few papers focus on selecting a suboptimal set of matchings to provide feedbacks to the learner. In particular, (De Souza et al. 2008) assesses overall structure of a CM using a graph isomorphism problem, however their algorithm is only for equal sized CMs.

Our contributions focus on improving feedbacks on local content by formalizing a *correlation feedback* for entity and proposition matchings, then on overall structure by selecting a suboptimal set of correlation feedbacks provided to the learner. Firstly, CM is defined as an *Ontological Concept Map* (OCM). A taxonomic and logical semantic repository based on a subset of OWL-lite ontology[1] defines entity semantics. The semantics is computed as corre-

[1] http://www.w3.org/TR/owl-ref/

lation feedbacks to match concepts or propositions. Correlation feedbacks are divided between visual representations on the learner OCM and semantic symbols converted into textual corrective comments visible on focus. Secondly, the set of correlation feedbacks provided to the learner is chosen through an overall assignment from the expert OCM to the learner OCM. A local search algorithm looks through all assignment solutions a suboptimal set of entity and proposition correlations. The quality of an assignment solution is computed by an objective function, based on entity and proposition similarities.

This paper is organized as follows. Section 2 formalizes expert and learner CMs as Ontological Concept Maps (OCM). Section 3 presents how the assessment, corresponding to an overall assignment of concepts and propositions from the expert OCM to the learner OCM, can be provided as a set of correlation feedbacks to the learner. Section 4 describes in detail correlation feedbacks respectively for entity and proposition matchings. Finally in section 5, the implementation of concept and proposition overall assignment is described as a local search algorithm using similarity measures to find a suboptimal set of correlation feedbacks.

2 Ontological Concept Map

CM is formalized as an Ontological Concept Map (OCM), based on an external ontology as a subset of OWL-lite language. The ontology specifies a constrained vocabulary on the CM topic and a description language. They are used to refine feedbacks with semantic correlations over CM concepts and relations. The language is formed by subsume hierarchy trees of classes and properties.

Definition 1 (ontology). Let Cl be a set of Classes, let Pr be a set of Properties. An *ontology* is a triple (Cl, Pr, Os), such as $Os = Os_{tax} \cup Os_{comp}$ with Os_{tax} (resp. Os_{comp}) the set of taxonomy (resp. comparison) statements:

- $Os_{tax} \subseteq (Cl \times \{subsume\} \times Cl) \cup (Pr \times \{subsume\} \times Pr)$
- $Os_{comp} \subseteq (Cl \times \{equivalent\} \times Cl) \cup (Pr \times \{equivalent, inverse\} \times Pr)$

Definition 2 (statement). Let $O = (Cl, Pr, Os)$ be an ontology. A *statement* is an element of Os.

Note that classes pf Cl and properties of Pr are organized in subsume hierarchy trees by Os_{tax}, starting from root node $\top$. Moreover, equivalent and inverse statements in Os_{comp} are considered only between nodes of the same depth in Os_{tax}.

Example 2.1. Examples are from an applicative learning course on Oriented Object Programming (OOP). Fig. 1 represents a part of taxonomy Os_{tax} and comparison Os_{comp} statements (e.g $(Class, subsume, SubClass)$, ...).

The OCM is based on the combination of ontological semantic and CM representation. Ontology semantics is transformed into CM by associating classes and their instances to concepts and properties to relations.

Definition 3 (Ontological Concept Map, proposition, starter and target concepts). Let $O = (Cl, Pr, Os)$ be an ontology, an *Ontological Concept Map* defined on O is an oriented labeled multigraph $M = (V, A, f_C, f_R)$, such as:

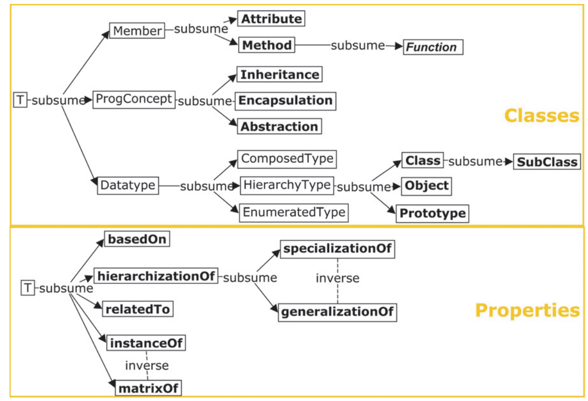

Figure 1: Part of the ontology on OOP

- V the set of vertices
- $A \subseteq V \times V$ the multiset of arcs
- $f_C : V \to Cl$ is the concept injective function
- $f_R : A \to Pr$ is the relation function

Let an arc a between vertices v_s, v_t $(v_s, v_t \in V)$. We call:

- $< f_C(v_s), f_R(a), f_C(v_t) >$ a *proposition* of M
- $f_C(v_s)$ (resp. $f_C(v_t)$) the *starter* (resp. *target*) concept

The set of propositions included within M is noted P_M.

Example 2.2. Fig. 2 shows a simplified version of the expert OCM on OOP. It contains a set of 12 vertices (labeled by concepts, e.g. $inheritance$, ...) and a set of 15 arcs (labeled by relations, e.g. $instanceOf$, ...) viewed as 15 propositions (e.g. $< subClass, specializationOf, class >$, ...)

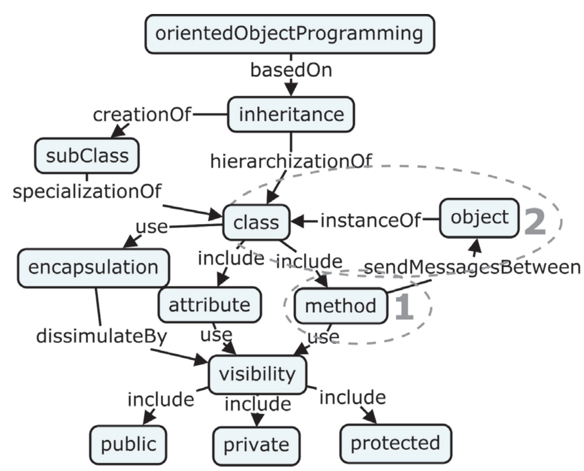

Figure 2: expert concept map on OOP

During course, the learner is asked to answer a focus question by creating its learner OCM. The ontology can be taken from ontology repositories or created specifically for the course. Learners have access to a restricted set of labels selected by the expert from the entities of the ontology (i.e classes and properties). The set of labels can contain distractors next to the question topic. In Fig. 1, classes and properties in bold are the labels given to the learner with *ones in italic being distractors*.

3 From assessment to correlation feedbacks

ITS produces feedbacks based on the assessment of a learner OCM. Feedbacks consist of visual elements drawn on the learner OCM and semantic corrective comments. The assessment is built on the assignment of concepts and propositions from the expert OCM with ones of the learner OCM. Each assignment is associated to a correlation feedback formalized as visual and semantic correlation feedbacks.

In this paper, assignment is limited to unique matchings such as each expert (resp. learner) concept or proposition is assigned to at most one learner (resp. expert) concept or proposition (def 4: A). Forcing unique assignment prevents missing or overnumbered feedbacks.

Definition 4 (overall assignment). Let an expert OCM $M_E = (V_E, A_E, f_{C_E}, f_{R_E})$ and a learner OCM $M_L = (V_L, A_L, f_{C_L}, f_{R_L})$ both defined on an ontology O. An *overall assignment* $A = (\pi_c, \pi_p)$ is a couple of assignment functions π_c and π_p assigning respectively concepts and propositions from M_E to M_L, such as $\pi_c : f_{C_E}(V_E) \rightarrow f_{C_L}(V_L)$ and $\pi_p : P_{M_E} \rightarrow P_{M_L}$ with π_c, π_p partial injective functions.

Each assignment of concepts π_c and propositions π_p is provided to the learner as a correlation feedback. It takes the form of a visual representation on the OCM learner and a symbol convertible to a corrective comment (visible on focus).

Definition 5 (correlation feedback *cor***).** Let an overall assignment $A = (\pi_c, \pi_p)$ from an expert OCM M_E to a learner OCM M_L. The *correlation feedback* $cor(a) = (\tau, \nu)$ is the feedback provided to the learner for an assignment pair $a \in A$, with τ the *visual correlation* and ν the *semantic correlation*.

The visual correlation τ is directly drawn on the learner OCM. Firstly, τ corresponds to the coloring of concept (resp. relation) labels from correct (in blue tone), partially correct (in gray tone) and wrong (in black tone) matchings. Arc arrows can be correct (in a straight line), partially correct (in a dotted line with gray tone) and wrong (in a dotted line with black tone). Second, τ can add a cross X on an arrowhead (resp. "INV" close to a relation label) to indicate that a learner arc is in the wrong direction (resp. the semantic inverse) in comparison to its assigned expert arc. Note that unassigned concepts and relations of learner and expert OCMs are also represented.

The semantic correlation ν is presented to the learner when he focuses on a vertex (resp. arc) of its OCM. ν is formalized as a symbol and presented as a corrective comment in association to the highlight of the assigned expert vertex (resp. arc).

Example 3.1. Fig. 3 represents visual correlations τ produced on a learner OCM. They are based on an overall assignment of the expert OCM (Fig. 2) on the learner OCM (Fig. 3). Following examples focus on the assignments a_1 and a_2 presented respectively in the gray frames [1] and [2].
$a_1 : \pi_c(method) = function$ matches learner concept $function$ with expert concept $method$. ν comments that $function$ is too specific ($(method, subsume, function)$ in Fig. 1) and τ colors $function$ in orange.
$a_2 : \pi_p(P_L) = P_E$ matches learner proposition $P_L =< class, redefinitionOf, prototype >$ and expert proposition $P_E =< object, instanceOf, class >$. ν indicates that learner arc is in the wrong direction (τ adds a cross on the arrownode). Moreover, ν indicates that learner has mistaken $redefinitionOf$ with $instanceOf$ (τ colors relation in gray) and $prototype$ with $object$ which are both related to $DataType$ (τ colors arrow in gray and as a dotted line)

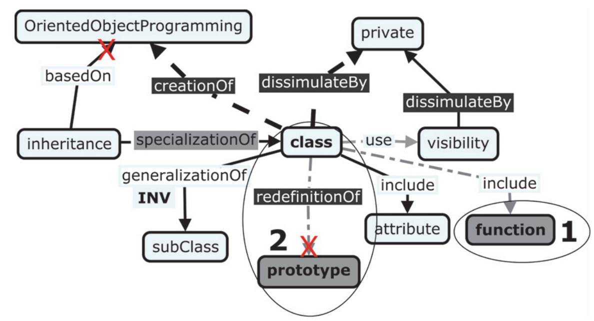

Figure 3: visual correlations on a learner concept map

Section 4 describes more precisely the range of correlation feedbacks that can be provided to the learner thanks to the ontological semantics.

4 Specifications of correlation feedbacks

Correlation feedbacks are divided between entity and proposition correlations. The *entity correlation* cor_e matches pairs of concepts or relations respectively from expert OCM and learner OCM, based on the external ontology. The *proposition correlation* cor_p matches pairs of propositions respectively from expert OCM and learner OCM, based on matchings of member pairs and the arc position.

Entity correlation feedback

The entity correlation feedback is defined as a couple $cor_e = (\tau_e, \nu_e)$. ν_e is a semantic symbol converted to a corrective comment. τ_e is a visual correlation changing the color of concept or relation labels.

Semantic ν_e and visual τ_e correlations indicate in the four simpler cases if an entity is similar (def 6: A) or equivalent (def 6: B) or subsumer (def 6: C) or subsumed (def 6: D) by an other entity. If the matching does not belong to these cases, ν_e provides their least common subsumer LCS (def 6: F). Note that if the $LCS(x, y)$ corresponds to the top node $\top$, x and y are considered semantically different (def 6: E).

Definition 6 (entity semantic correlation ν_e**).** Let an ontology $O = (Cl, Pr, Os_{tax} \cup Os_{comp})$, let $x, y \in (Cl \times Cl) \cup (Pr \times Pr)$ be a pair of entities. The *entity correlation*

$\nu_e(x, y)$ is defined as:

(A)	$x = y$	if $x = y$
(B)	$x \equiv y$	if $(x, equivalent, y) \in Os_{comp}$
(C)	$x \sqsupseteq y$	if $[(x, subsume..., y)] \in Os_{tax}$
(D)	$x \sqsubseteq y$	if $[(y, subsume..., x)] \in Os_{tax}$
(E)	$x \neq y$	if $LCS(x, y) = \top$
(F)	$LCS(x, y)$	otherwise

Semantic symbols ν_e are converted into semantic comment using the following treatment: "Your [concept|relation] y is {equivalent (A,B), too specific (C), too general (D), should be replaced (E), is related by $LCS(x, y)$ (F)} [by|in comparison to] the expert [concept|relation] x".

Example 4.1. In fig. 3, the learner is provided with visual and semantic feedbacks. For [1], the visual correlation $\nu_e(function, method)$ is colored in gray tone and the semantic comment "Your concept $function$ is too specific in comparison to the expert concept $method$" (translated from the symbol $function \sqsupseteq method$) appears on learner's focus.

Proposition correlation feedback

CM representation is composed of a set of propositions, each one describing an unique knowledge sentence. Proposition correlation is based on the matchings of member pairs and the position type in which the relation is considered: normal or reversed. The normal position matches starter to starter and target to target concepts whereas the reversed position matches starter (resp. target) concepts from a proposition to target (resp. starter) concepts of the other proposition.

The proposition correlation cor_p is provided as feedback and is divided in three types: *direct* $cor_{p_{dir}}$ with a normal position, *wrong direction* $cor_{p_{wd}}$ and *inverse relation* $cor_{p_{inv}}$ with reversed positions. First, the direct correlation $cor_{p_{dir}}$ corresponds to the simplest matching case, where propositions are semantically close. Second, the wrong direction correlation $cor_{p_{wd}}$ focuses on propositions that are semantically close but where the learner has written the relation in the opposite direction. Third, the inverse relation correlation $cor_{p_{inv}}$ considers a pair of propositions with in opposite direction of arrow and relations which are inverse. The proposition correlation $cor_p = (\tau_p, \nu_p)$ is divided between the semantic correlation ν_p and the visual correlation τ_p.

The semantic correlation ν_p corresponds to a quadruplet of semantic symbols. ν_p indicates which correlation type corresponds the most to the proposition matching and each entity semantic correlation between proposition members, depending of the matching position (i.e normal or reversed).

Definition 7 (**proposition semantic correlation** ν_p)**.** Let $p_E =< c_E, r_E, c'_E >$ (resp. $p_L =< c_L, r_L, c'_L >$) be a pair of propositions within expert OCM (resp. learner OCM). The *proposition semantic correlation* ν_p between p_L and p_E is a quadruplet defined on normal matching position:

$$(cor_{p_{dir}}, \nu_e(c_L, c_E), \nu_e(r_L, r_E), \nu_e(c'_L, c'_E))$$

or reversed matching position:

$$(\{cor_{p_{wd}}, cor_{p_{inv}}\}, \nu_e(c_L, c'_E), \nu_e(r_L, r_E), \nu_e(c'_L, c_E))$$

Semantic symbols ν_p are converted into semantic comments using the following treatment. The first line describes the correlation type: "Your proposition p_L {is directly matched, has its arrow in the wrong direction, is semantically the inverse} in comparison to the expert proposition p_E". The three other lines convert semantic symbols ν_e from member matchings (cf. section 4).

The visual correlation τ_p colors in a first phase the relation label and the arc arrow. Relation label coloring is based on τ_e. Arc arrow coloring depends of both matchings between concept members: dashed line with black tone if at least one is a wrong matching (ν_e equals to $\neq$), otherwise dashed line with gray tone if at least one is a partially correct matching ($\nu_e \in \{\sqsupseteq, \sqsubseteq, LCS\}$) and otherwise straight line with black tone when matchings are equivalent or equal ($\nu_e \in \{=, \equiv\}$). In a second phase for reversed position matchings, a cross X is drawn on the arrowhead for wrong direction matching ($cor_{p_{wa}}$) whereas an "INV" label is written close to the relation label for inverse relation matching ($cor_{p_{inv}}$).

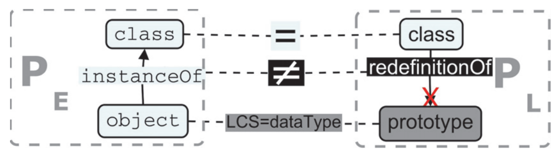

Figure 4: correlation feedback for proposition matching

Example 4.2. Between the expert proposition P_E and the learner proposition P_L in Fig. 4, the visual correlation $\nu_p(P_E, P_L)$ can be considered as direct, wrong direction or inverse relation correlations (Fig. 4). The matching corresponds to the wrong direction correlation $cor_{p_{wd}}$ translated into the following corrective comment. "Your proposition p_L has its arrow in the wrong direction in comparison to the expert proposition p_E. Your relation $redefinitionOf$ should be replaced by the expert relation $instanceOf$. Your concept $prototype$ is related by $dataType$ to the expert concept $object$".

Finally section 5 gives details on how feedbacks are computed and assigned from expert OCM to learner OCM.

5 Overall assignment of correlation feedback

In CM ITS, the assessment consists of maximizing the quality of concept and proposition assignments between the expert CM and the learner CM. Similarity measures formalize the matching distance between a pair of entities (as *entity similarity* δ_e) or propositions (as *proposition similarity* δ_p). Similarities are numeric values between $[0, 1]$ with 0 the best matching.

Thus, the assessment can be related to an assignment problem searching the overall assignment solution of minimum similarity cost. This cost is computed by an objective function and can be presented as the learner score. Since expert and learner CMs can have different sizes of concept or proposition sets, the assessment corresponds to a NP hard problem and therefore this paper searches an approximate assignment solution as suboptimal set of correlation feedbacks.

Entity similarity

The *entity similarity* δ_e is computed based on ontological semantic measures. This paper adapts an information theoretic measure (Al-Mubaid and Nguyen 2006) by defining the similarity correlation of an entity pair based on:

- their knowledge closeness (i.e distance of the *minimum statement path* $minP$ separating them in the ontology)
- the correlation informativity (i.e the *common specificity* $Cspec$ corresponding to the depth of their least common subsumer (LCS) in the subsume hierarchy tree)

This paper adds refinments to δ_e based on CM features. First, entity pairs having for LCS the top node $\top$ are different (i.e $\delta_e = 1$). Moreover, $minP$ can contain at most one inverse statement (relations will be inverse) and if $minP$ contains only an equivalent statement, the entity pair is equivalent (i.e $\delta_e = 0$).

Proposition similarity

The *proposition similarity* δ_p measures the closeness of a pair of propositions by summing values of each similarity computed between pairs of proposition members. Similarities δ_e for pairs of members are based on the correlation type (cf. section 4) in which the relation is considered (i.e direct, wrong direction and inverse relation). The similarity δ_p corresponds to the correlation type maximizing the feedback quality (i.e minimizing $delta_p$). In case of several best correlation types, δ_p is chosen based on following order: $\delta_{p_{dir}} > \delta_{p_{wd}} > \delta_{p_{inv}}$.

Definition 8 (proposition similarity δ_p). Let p_E $=< c_E, r_E, c'_E >$ (resp. $p_L =< c_L, r_L, c'_L >$) be a pair of propositions within expert OCM (resp. learner OCM). Proposition similarities for *direct matching* $\delta_{p_{dir}}$, *wrong direction* $\delta_{p_{wd}}$ and *inverse relation* $\delta_{p_{inv}}$ are defined as:

$$\begin{aligned}
\delta_{p_{dir}} &= \alpha.\, \delta_e(c_E, c_L) + \beta.\, rel(r_L, r_E) \quad + \gamma.\, \delta_e(c'_E, c'_L)\\
\delta_{p_{wd}} &= \alpha.\, \delta_e(c'_E, c_L) + \beta.\, \theta(rel(r_L, r_E)) + \gamma.\, \delta_e(c_E, c'_L)\\
\delta_{p_{inv}} &= \alpha.\, \delta_e(c'_E, c_L) + \beta.\, rel_{inv}(r_L, r_E) + \gamma.\, \delta_e(c_E, c'_L)
\end{aligned}$$

with:

- $\alpha, \beta, \gamma \in]0, 1[$ normalizing constants, $\alpha + \beta + \gamma = 1$
- $\theta : [0, 1] \rightarrow [0, 1]$ the malus function such as $\theta(x) \geq x^2$
- $rel(r_L, r_E)$ (resp $rel_{inv}(r_L, r_E)$) relation functions equals to
  - 1 if the minimum path $minP(r_L, r_E)$ contains (resp. does not contain) an inverse statement
  - $\delta_e(r_L, r_E)$ otherwise

The *proposition similarity* δ_p between p_L, p_E is defined as:

$$\min\{\delta_{p_{dir}}(p_L, p_E), \delta_{p_{wd}}(p_L, p_E), \delta_{p_{inv}}(p_L, p_E)\}$$

Example 5.1. In Fig. 5, the learner proposition p_L and the expert proposition p_E are matched. The minimum similarity of the three matching types is $\delta_{p_{wd}} = \frac{0}{3} + \frac{1}{3} + \frac{0.3}{3} = 0.43$. So $\delta_p(p_L, p_E) = \delta_{p_{wd}} = 0.43$

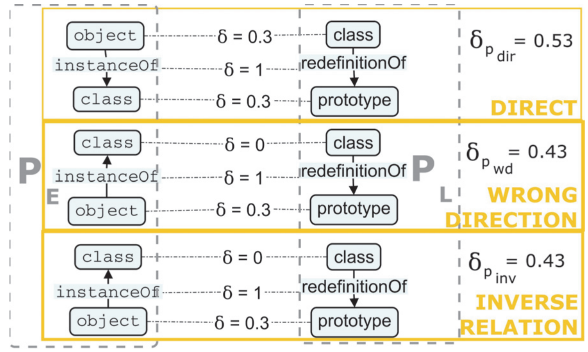

Figure 5: Similarity types for proposition matching

The overall assignment is based on similarities δ_e between concepts and δ_p between propositions, precomputed to be used during the assessment.

Overall assignment

First, the objective function fit is defined to measure the quality of an assignment solution $A = (\pi_c, \pi_p)$ of concepts and propositions. fit sums similarities of each assigned concept δ_e (def 9: A) and assigned proposition δ_p (def 9: B). These sums are divided by the number of concepts (resp. propositions) within the expert OCM. Then malus are added to the objective function as the ratio of number of unassigned concepts (def 9: C) and propositions (def 9: D) on the total number of concepts (resp. propositions) within both expert OCM and learner OCM.

Definition 9 (objective function fit). Let π_p, π_c an assignment of an expert OCM $M_E = (V_E, A_E, f_{C_E}, f_{R_E})$ on a learner OCM $M_L = (V_L, A_L, f_{C_L}, f_{R_L})$. The *objective function* $fit(\pi_c, \pi_p)$ is defined such as:

$$\begin{aligned}
&(A) \qquad \Upsilon * \frac{\sum_{c_i \in f_{C_E}(V_E)} \delta_e(c_i, \pi_c(c_i))}{|V_E|}\\
&(B) \;+\; \Omega * \frac{\sum_{p_i \in P_{M_E}} \delta_p(p_i, \pi_p(p_i))}{|P_{M_E}|}\\
&(C) \;+\; \Psi * \frac{|V_L| + |V_E| - 2 * |\pi_c|}{|V_L| + |V_E|}\\
&(D) \;+\; \Phi * \frac{|P_{M_L}| + |P_{M_E}| - 2 * |\pi_p|}{|P_{M_L}| + |P_{M_E}|}
\end{aligned}$$

with $\Upsilon, \Omega, \Psi, \Phi \in]0, 1[$ be normalizing constants, such as $\Upsilon + \Omega + \Psi + \Phi = 1$

Second, finding a suboptimal approximate assignment solution is based on an iterated local search algorithm divided into a repeated two-phase process. The *hill climber* permutates assignment of two expert concepts at each move, followed by a random relaunch when search stops into a local optimum. Note that to prevent approximative assignments to take the place of exact assignments, pair of concepts (resp. propositions) having the same labels are always assigned together. During first phase, the hill climber (HC) algorithm

[2] we chose $\alpha = \beta = \gamma = 1/3$ and $\theta(x) = \sqrt{x}$

searches through assignment solutions a suboptimal assignment solution. HC is a Local Search technique, i.e an iterative search procedure called intensification which starts from an initial solution and progressively improves it by applying a series of local moves. In this paper, at each iteration, the HC reaches an one-move neighbor solution by permuting assignment of two expert concepts[3]. The HC stops when it encounters a local optimum, a solution which has no improving neighbors. The second phase starts when the search is blocked in a local optimum. Therefore, a random relaunch diversifies solution pool by starting at a new random initial solution.

Discussion

ITS assessment has been used in CM softwares such as CMapTools, IKAS or COMPASS. In relation to this paper, a new ITS software called IOCMap (Intelligent Ontological Concept Map) has been created and used by thirty sophomore college students learning the computer science. Students had access to this software during practical work and they created their own CM representing their knowledge on object oriented programming, based on a set of concept and relation labels. Learners received a complete set of correlation feedbacks (as presented in this paper).

The set of computed feedbacks has been compared with manual feedbacks assessed on the learner CMs. The implementation assigns consistantly correlation feedbacks for learner CMs. Feedbacks are not partial, indeed there are neither any missing nor overnumbered expert assignment(s) for each learner concept and proposition. It is due to the uniqueness of concept and proposition assignments between expert and learner CMs. However, our algorithm has been limited by the size of the external ontology. Indeed OOP is centered around key terms that are defined in a strict and limited semantics, thus corrective comments are mostly assessing correctness or wrongness whereas rarely explaining partially correct answers of the learner.

In CM ITS, few papers use AI techniques to compute automatically an overall score of a learner CM. Not any paper at our knowledge focused on finding a set of feedbacks based on graph matching techniques. Lack of formalism in cognitive science papers and the flexibility in CM semantics seem to create an unlikely marriage between these fields. This work is intended to show that both communities could benefit from combining their knowledge. Assessment can benefit from the overall assignment of feedbacks between expert and learner CMs. The need for measures to match entities and propositions based on an external semantics does not imply mandatory losses in CM flexibility. Indeed, external semantics can be used only as a disembiguation process. One could imagine systems based on wider external semantics (e.g wordnet), taking into account incomplete semantics within their computation.

[3]proposition assignments are computed by the Munkres algorithm. It selects unique proposition matchings (i.e unique matching assigned by row and column) such as the sum of proposition similarities is the lowest. It is refined so that matchings which cannot be assigned depending of concept assignments (i.e learner proposition having starter and target concepts assigned to the ones of the expert proposition) will not be selected at first

References

Al-Mubaid, H., and Nguyen, H. 2006. A cluster-based approach for semantic similarity in the biomedical domain. In *Engineering in Medicine and Biology Society, 28th Annual International Conference of the IEEE*, 2713–2717.

Alves da Silva, A.; Padilha, N.; Siqueira, S.; Baio, F.; and Revoredo, K. 2012. Using concept maps and ontology alignment for learning assessment. *IEEE Technology and Engineering Education* 7.

Canas, A.; Novak, J.; and Reiska, P. 2015. How good is my concept map? am i a good cmapper? In *Knowledge Management & E-Learning*.

Cline, B. E.; Brewster, C. C.; and Fell, R. D. 2010. A rule-based system for automatically evaluating student concept maps. *Expert Systems with Applications* 37(3):2282–2291.

Da Rocha, F.; Favero, E.; and Da Costa Junior, J. 2008. An approach to computer-aided learning assessment. In *3rd Concept Map Conference*, volume 1, 100–107.

De Souza, F.; Boeres, M.; Cury, D.; De Menezes, C.; and Carlesso, G. 2008. An approach to comparison of concept maps represented by graphs. *3rd Concept Map Conference*.

Gouli, E.; Gogoulou, A.; Papanikolaou, K.; and Grigoriadou, M. 2005. Evaluating learner's knowledge level on concept mapping tasks. In *ICALT*, 424–428.

Graudina, V.; Grundspenkis, J.; and Milasevica, S. 2012. Ontology merging in the context of concept maps. In *Applied Computer Systems*, 29–36.

Harrison, S. H.; Wallace, J. L.; Ebert-may, D.; and Luckie, D. 2004. C-tools automated grading for online concept maps works well with a little help from wordnet.

Kharatmal, M., and Nagarjuna, G. 2006. A proposal to refine concept mapping for effective science learning. In *2nd Concept Map Conference*.

Limongelli, C.; Lombardi, M.; Marani, A.; Sciarrone, F.; and Temperini, M. 2016. Concept maps similarity measures for educational applications. In *13th Int. Conf. on Intelligent Tutoring Systems (ITS)*.

Marshall, B.; Chen, H.; and Madhusudan, T. 2006. Matching knowledge elements in concept maps using a similarity flooding algorithm. *Decision Support Systems*.

McClure, J.; Sonak, B.; and Suen, H. 1999. Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Research in Science Teaching* 36(4).

Novak, J., and Gowin, D. 1984. *Learning how to learn*. Cambridge University Press.

Park, U., and Calvo, R. 2008. Automatic concept map scoring framework using the semantic web technologies. In *ICALT*.

Schwendimann, B. 2011. *Mapping biological ideas: Concept maps as knowledge integration tools for evolution education*. Ph.D. Dissertation, UC Berkeley.

Soika, K., and Reiska, P. 2014. Assessing students' interdisciplinary approach with concept mapping. In *5th Concept Map Conference*.

Strautmane, M. 2012. Concept map based knowledge assessment tasks and their scoring criteria : an overview. In *5th Concept Map Conference*.