ALES: An innovative argument-learning environment

Safia Abbas[1], Hajime Sawamura[2]

(1. Graduate School of Science and Technology, Niigata University, Niigata 950-2181, Japan;

2. Institute of Natural Science and Technology, Niigata University, Niigata 950-2181, Japan)

Abstract: This paper presents the development of an Argument-Learning System (ALES). The idea is based on the AIF (argumentation interchange format) ontology using "Walton theory". ALES uses different mining techniques to manage a highly structured arguments repository. This repository was designed, developed and implemented by the authors. The aim is to extend the previous framework proposed by developing an intelligent tutoring environment for argument learning that aims to: (1) guide the students during argument learning; (2) aid in improving the students' argument skills; and (3) offer an argument classifier agent that retrieves the most relevant results to the subject of search. This paper focuses on the environment development specifying the status of each of the constituent modules.

Key words: argumentation; mining techniques; intelligent tutoring system

1. Introduction

Argumentation theory is considered as an interdisciplinary research area. Its techniques and results have found a wide range of applications in both theoretical and practical branches of artificial intelligence and computer science (Baker, Andriessen & Suthers, 2003; Rahawan & Sakeer, 2006; Reed & Rowe, 2003). Recently, AI (argumentation interchange) in education is interested in developing instructional systems that help students hone their argumentation skill (Aleven & Ashley, 1997). This skill is extremely valuable in the educational field, and it reflects the students' abilities to outline a claim in a logical and convincing way and provides supportable reasons for the claim as well as identifies the often implicit assumptions that underlie the claim. Although argumentation skill is very important in the field of education, students' main barrier is their inabilities to follow the argument which highlights the main points of a context (Harrell, 2006). In response to the importance of argumentation skills in education, different argument mapping tools (e.g., Compendium, Araucaria, Rationale, etc.) have been developed (Reed & Rowe, 2003). These tools are designed to foster students' abilities to articulate, comprehend and communicate reasoning by producing diagrams of reasoning and argumentation. The main drawback in these tools is the absence of an administrator to constrict the argument diagram process, in other words, guiding the students to analyze arguments based on scientific theories or evidence (Paolucc, Suthers & Weiner, 1996).

In this paper, the authors extend previous framework (Abbas & Sawamura, 2008b) proposed by developing an intelligent learning environment ALES that uses mining agent-based ITS (intelligent tutoring system) for teaching in argument. ALES uses the highly structured argument repository "RADB" (relational argument database) to expose expert knowledge. It also models the students' argumentation knowledge and skills then, based on this information, it presents a group of arguments from which the users can choose one to work on. The

Safia Abbas, Ph.D. candidate, Graduate School of Science and Technology, Niigata University; research field: computer science.
Hajime Sawamura, professor, Institute of Natural Science and Technology, Niigata University; research field: computer science.

paper is organized as follows: Section 2 introduces the learning environment (ALES) architecture. Section 3 shows an illustrative example. Related work and discussion are presented in section 4. Finally, conclusion and future work are illustrated in section 5.

2. ALES architecture

This section describes the architecture of the proposed learning environment (ALES) as shown in Figure 1. The environment consists of 4 main parts: (1) The domain model is represented as a highly structured argument repertoire; (2) The pedagogical model contains 3 components: a parser, a classifier agent and a teaching model; (3) The student model keeps track of the student performance and assists the pedagogical model in offering the individualized teaching; and (4) Finally, the interface model is "GUI" (graphical user interface). Not only does ALES teach argument analysis, but also assesses the students and guides them through personalized feedback. The next subsections illustrate the domain and pedagogical models in details, whereas the student-teaching model interaction is out of the scope of this paper.

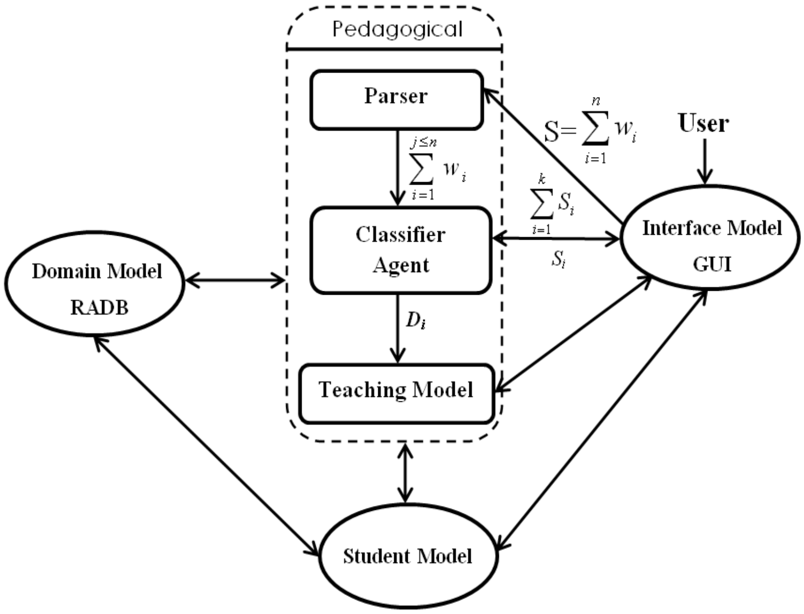


Figure 1 ALES architecture

2.1 The domain model

The domain model is represented in the form of the RADB, it has been developed and implemented by the authors (Abbas & Sawamura, 2008b; 2008c) which summon a huge number of arguments. These arguments were previously analyzed by experts based on Walton theory using the AIF ontology (Rahawan & Reed, 2007; Zablith, Rahawan & Reed, 2007). The domain model can semantically be represented as a forest of a numerous directed free trees (Chi & Muntz, 2001). Each directed tree in the forest lays out a semantic representation for a specific argument analysis. The domain model representation is general enough to encapsulate multiple domains, it also enjoys the extendibility feature, where adding new schemes are permitted. Figure 2 describes the various building blocks concerned with the RADB, using screen shots of the authors' implemented system, such as: (1) the table "Scheme_TBL" gathers the name and the index for different schemes; (2) the table "Scheme_Struct_TBL" assembles the details of each scheme in "Scheme_TBL"; and (3) the "Data_TBL" table contains the analysis of different arguments based on different scheme structures and preserves the constraints of the AIF ontology (Rahawan & Reed, 2007).

Scheme_TBL : Table

ID	SCH_Name
1	Expert Opinion
2	Popular Opinion
3	Verbal Classific
5	inference
6	Conflict
7	Prefrence
8	Depate Result

(a)

Scheme_Struct_TBL : Table

ID	SCH_ID	Type	Content
1	1	P	Source E is an expert in the subject domain X containing proposition B.
2	1	P	E asserts that proposition B in domain X is true.
3	1	C	B may plausibly be taken to be true.
4	8	CC	Critical argumentation conclusion
5	1	CQ	Expertise Question: How credible is expert E as an expert source?
6	1	CQ	Field Question: Is E an expert in the field that the B is in?
7	1	CQ	Opinion Question: Does E's assertions imply B?
8	1	CQ	Trustworthiness Question: Is E reliable as source?
9	1	CQ	Consistency Question: Does B consistent with the assertions of other experts?
10	1	CQ	Backup Evidence Question: are there any evidences sustain B?
11	5	RA	RA-Node
12	6	CA	CA-Node
13	7	PA	PA-Node

(b)

Data_TBL : Table

ID	Stru_ID	Content	Type	Child_of	level	argumentation_no
1	3	Kyle Mutch's tragic death was not an accident and he suffered injuries	-1	0	0	argument_602
2	7	RA-Node	1	1	1	argument_602
3	2	Dr.James told the high court at Forfar the youngest death could not be cau	1	2	2	argument_602
4	1	Dr.James Grieveis a Pathologists who examine the baby shortly after his (	1	2	2	argument_602
5	4	Death, not an accident, court told	1	2	2	argument_602
6	7	RA-Node	1	5	3	argument_602
7	6	Field Question: Dr.James Grieve is an expert in pathology	1	6	4	argument_602
8	7	Opinion Question: the baby is a child that is not walking,	1	6	4	argument_602
9	10	Backup Evidence Question: the lecturer in forensic medicines	1	6	4	argument_602
10	12	CA-Node	0	5	3	argument_602
11	9	conflict from inconsistent testimony: the baby's stepfather denies murder	0	10	4	argument_602

(c)

Figure 2 The main tables in RADB

2.2 Pedagogical model

The pedagogical model is responsible for reasoning about the students' behaviors according to the students' model, in order to: (1) retrieve the most relevant results to both the subject of search and the students' background; (2) expose the corresponding argument to the selected result; and (3) guide the students' analysis based on the pre-existing one. The pedagogical model as seen in Figure 1 consists of three main components: a parser, a classifier agent and a teaching model.

2.2.1 Parser

The parser receives a statement "S" from the students. This statement is divided by the parser into tokens, and then, the number of tokens is reduced. Finally, the final crucial set of words {w_1 w_2... w_n} is sent to the classifier agent. The tokens are reduced if they belong to a look-up table containing a set of all unnecessary words like {a, an, the, he, have, is, him ...}, otherwise, they are added to the set of tokens to be sent to the classifier agent. The importance of the parser module lies in reducing the set of tokens into a set of significant keywords, which in turn will: (1) improve the results of the classifier where combinations of unnecessary words vanish; and (2) reduce the number of iterations done by the classifier agent. The parser has already been implemented and discussed (Abbas & Sawamura, 2008c).

2.2.2 Classifier agent

The classifier agent gathers and controls different mining techniques in order to classify the retrieved contexts based on students' choices. The agent mines the RADB repository aiming to: (1) direct the search process towards hypotheses that are more relevant to students' subject of search, classify the analogous arguments in different ways based on students' choices and seek for the most relevant arguments to the subject of search; and (2) add flexibility to the retrieving process by offering different search techniques. The agent offers 3 search techniques: priority search, rule extraction search and polarity search. In the former, the priority search classifies and retrieves contexts based on the maximum support number using an adapted version of the AprioriTid

(Agrawal & Muntz, 1994) mining technique. The rule extraction summarizes the retrieved arguments searching for hidden patterns that are most relevant to the subject of search, and then, these patterns are exposed in the form of rules. Each rule, for each retrieved argument, contains the affirmative "+" and the negative "-" parts relating to the final conclusion of that argument. In the latter, the polarity search classifies the retrieved arguments into 2 classes: affirmative and negative, relevant to the subject of search and based on the search criteria. The search criteria depends on the students' choices, it can be either by premises or by conclusion. The premises (with/against) criterion retrieves arguments by searching only in the different premises. Similarly, conclusion retrieves the arguments by searching only in the different conclusions.

(1) Priority search

The AprioriTid algorithm (Agrawal & Muntz, 1994) has been implemented and embedded to the classifier agent as "priority search" as seen in Figure 3. The priority search aims to retrieve the most relevant arguments to the users' subject of search and queue them based on the maximum support number, so that the first queued argument is the one that has more itemsets (Agrawal & Muntz, 1994) related to the subject of search. Although the AprioriTid algorithm has originally been devised to discover all significant association rules between items in large database transactions, the agent employs its mechanism in the priority search to generate different combinations between different itemsets (Abbas & Sawamura, 2008c; Agrawal & Muntz, 1994). These combinations will then be used to classify the retrieved contexts and be queued in a descending order based on their support numbers. As a response to the priority search purpose, an adapted version of the AprioriTid mining algorithm has been applied. This adapted version, as seen in Figure 4, considers the single itemset (1-itemset) size as well as the maximum support number usage, rather than k-itemset for $k \geq 2$ and the minimum support number "minsup" mechanism. For more clarification, the priority search mines specific parts of the pre-existing arguments based on the users' search criteria. These search criteria enable the students to seek the premises, conclusions or the critical questions lying in the different arguments. For example, suppose a student queries the RADB searching for all information related to "Islamic inheritance rules". Simply, he/she may write "the inheritance regularities in Islam" as the search statement and can choose the conclusion as the search criteria. In this case, the classifier agent receives the set of significant tokens {inheritance, regularities, Islam} from the parser model. This set is considered as the single size itemset (1-itemset) $C_1=\{w_1, w_2, w_3\}$ that contains the most crucial set of words in the search statement.

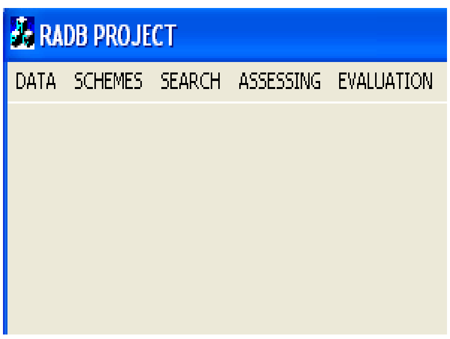


Figure 3 The main window of the implemented system in Visual C++

1) $L_1 = \{large\ 1\text{-}itemsets\};$
2) For each itemset $C_1 \in L_1$ *repeat steps 5 and 6 for k=1;*
3) For $(k=2;\ L_{k-1} \neq \Phi;\ K^{++})$ *do begin*
4) $C_k = apriori\text{-}gen(L_{k-1});$ *//New candidates*
5) For all transactions $t \in D$ *do begin*
6) $\overline{C}_k = subset\ (C_k, t);$ *//candidates contained in t*
7) end;
8) For all similar candidates $c \in U_k\ \overline{C}_k$ *do*
9) $c.count^{++};$ *//the support number*
10) $Ans = \{c \in \overline{C}_k \mid descending\ ordered\};$

Figure 4 An enhanced version of AprioriTid

Then, the agent uses the adapted version of the AprioriTid algorithm to generate the different super itemsets $C_{\{2 \leq k \leq 3\}}$, which are the different combinations between different tokens. Therefore, the generated super itemsets, as seen in Figure 5, will be the 2-itemset $C_2=\{w_1w_2, w_1w_3, w_2w_3\}$, and the 3-itemset $C_3=\{w_1w_2w_3\}$. Afterward, the different conclusions in the different arguments trees will be mined seeking for the most relevant set of arguments Ans={d_1, d_2, ...,d_m} so that $\forall d_i \in$ Ans $\exists C_{k\in \{1,2,..,j\}} \subseteq d_i$}. Finally, the results will be queued in a descending order and exposed in a list, where the students can choose the argument name “Argument_314” from the list to expose the associated context and analysis.

$$D = \{(1, Context_1,\ argument\_602), (2, Context_2,\ argument\_314),\ \ldots, etc.\}.$$

L_1

C_1	$\overline{C}_1$
w_1	{1,2,7}
w_2	{1,3,5}
w_3	{1,2}

L_2

C_2	$\overline{C}_2$
{w_1 w_2}	{1,2}
{w_1 w_3}	{1,2}
{w_2 w_3}	{1}

L_3

C_3	$\overline{C}_3$
{w_1 w_2 w_3}	{1}

Ans

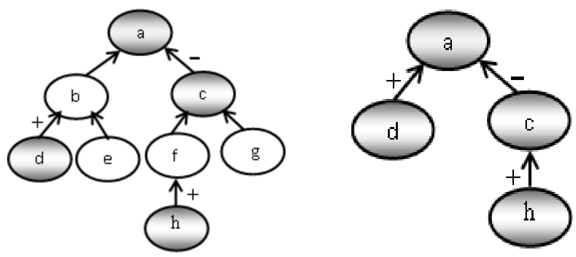
Argument ID	Support number
1	7
2	4
7	1
3	1

C_k	Set of candidate k-itemsets
$\overline{C}_k$	The ID transactions associated with each candidate
L_K	Set of large k-itemsets. Each member of this set has two fields (i) C_K (ii) $\overline{C}_K$
D	The transactions “Data_TBL” query that contains three fields: (i) transaction ID (ii) the associated content/context (iii) the associated argument_no.

Figure 5 The adapted AprioriTid mechanism

(2) Rule extraction search

Rule extraction mining is a search technique in which argument trees are encountered to discover all hidden patterns “embedded subtrees” (Chi & Muntz, 2001) that coincide with the relation between some objects. These objects express a set of the most significant tokens of the users’ subject of search. Precisely, it is supposed that a student wants to report some information about the relation between the “USA war” and the “weapons of mass destructive”. At the beginning, the users’ search statements are reduced to the most significant set of tokens by the parser (Abbas & Sawamura, 2008a; 2008b; 2008c). Then, the different argument trees, pre-existing in the RADB repository, are mined in order to fetch these different tokens.



(a) (b)

Figure 6 The tree rule extraction search

Figure 6a shows the analysis of an argument tree, where some enclosed nodes coincide with the students' search statements, while Figure 6b shows the revealed embedded subtree. Finally, each resulted subtree is expressed in the form of a rule as shown in Figure 7, where "+" indicates that this node is a support to the final conclusion, whereas "-" is a rebuttal node to the final conclusion.

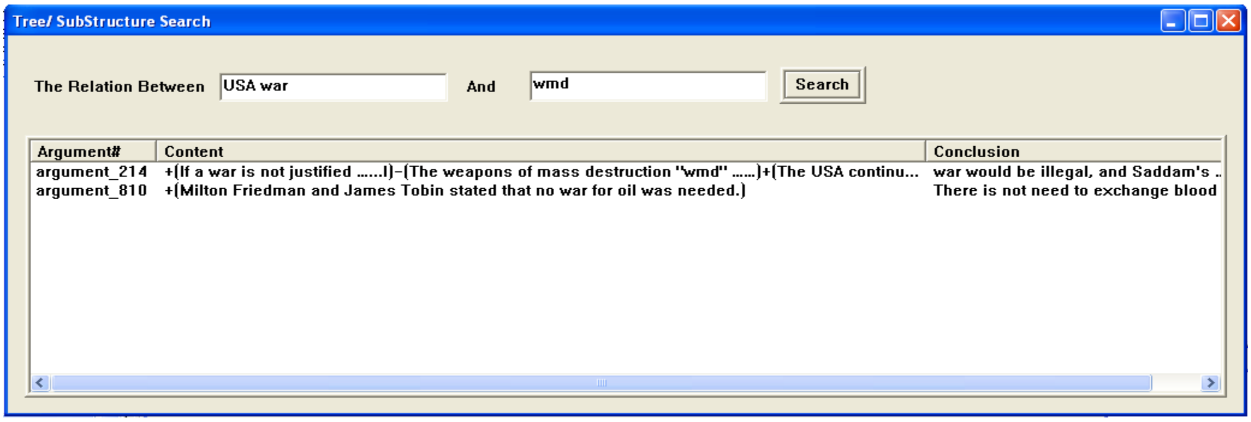


Figure 7 The representation form of rule extraction search result

2.2.3 Teaching model

The teaching model monitors the students' actions, guides the learning process and provides the appropriate feedback. However, in the meantime, it is still in the implementation phase. The model starts its role when the classifier agent sends the document D_i selected by the students. The teaching model checks, according to the current student model, whether the students are in the learning or the assessing phase. If the students are in the learning phase, the document is presented associated with the corresponding analysis. On the other hand, if the students are in the assessment phase, they are able to do their own analysis, and the teaching model will guide them during analysis by providing personalized feedback whenever required. The feedback aims to guide the students and refine their analysis and intellectual skills. Two kinds of feedback are provided by the teaching model: partial argument negotiation and total argument negotiation.

(1) Partial negotiation

In this case, the students start analyzing the argument context in the form of a tree in which the root holds the final conclusion of the issue of discussion. The teaching pedagogy used in this case provides partial hints at each node of the analysis tree. They are results of comparing the students' current node analysis to the original one in the argument database. These hints are provided before allowing the students to proceed further in the analysis process, which aim to minimize the analysis error ratio as much as possible, for the current analyzed node. Generally, the teaching model guides the students via the partial hints at each node till the error of the current node is minimized to a specific ratio. After then, the students are able to move to the next analysis step (i.e., node).

(2) Total negotiation

The total negotiation is similar to the partial negotiation. However, the teaching pedagogy is different in that it provides hints only at the end of the analysis process. In other words, after the students build the full analysis tree for the selected context, the system interprets and evaluates the students' analysis comparable to the pre-existing one and remarks the errors.

Generally, in the assessing phase, the teaching model presents the transcript of the chosen argument associated with an empty tree skeleton and asks the students to start their own analysis. The students start the analysis by copy and paste text passages from the transcript or freely enter text into the nodes. The teaching model traces each node text and divides it into set of significant tokens, then interprets and evaluates the errors ratios comparable to the pre-existing analysis underlying in the RABD. Finally, the model provides the feedback, partially or totally, based on the students' choices and records the students' errors for the current transcript, which in turn will be used, by the student model, to evaluate the performance and to follow the progress of the students.

2.3 The student model

The student model stores details about students' current problem-solving state and long-term knowledge progress, that is essential for students' future performance evaluations. The model considers personal information, pre-test evaluation and performance history. Personal information contains personal data as name, ID, password, state (learning/assessment), feedback_type (partial/total) …. The pre-test evaluation permanently assesses the students' argument analysis skills and follows the student progress through learning process. Finally, the performance history implicitly reflects how much the students have done and how well.

3. An illustrative example

This example shows a complete run for partial negotiation of the assessing phase. The system interactions are written in normal font. The users' actions are in bold. The authors' illustrations to some actions will be italicized.

Supposing the students in the assessing phase choosing the partial feedback property, the system will give the students the ability to select specific scheme to be used in their analysis, as shown in Figure 8.

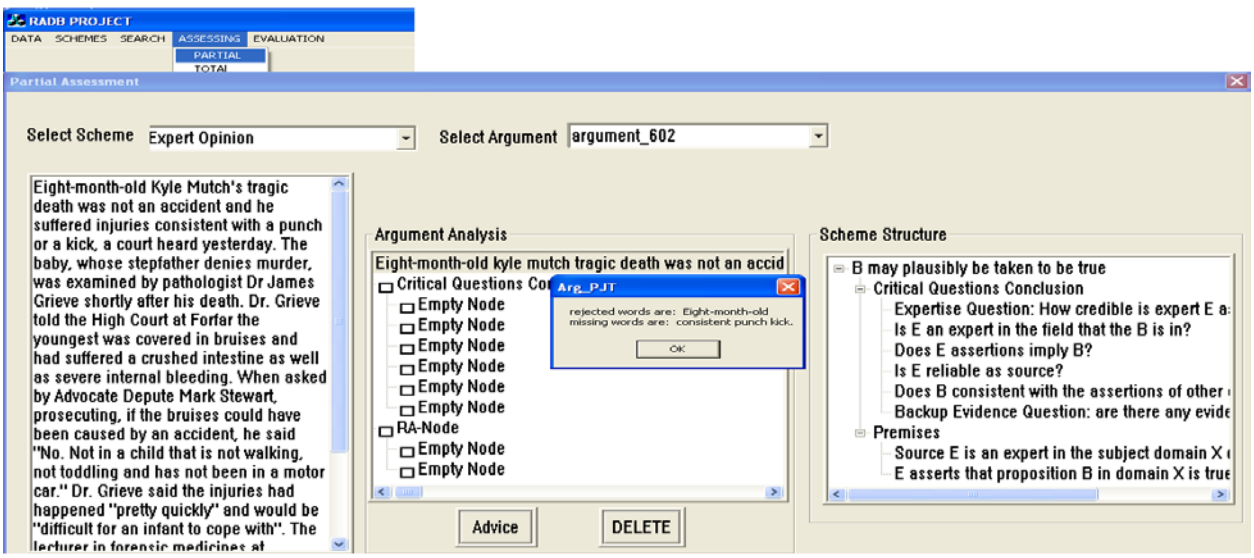


Figure 8 The partial assessment form

User—"expert opinion scheme".

The whole arguments, that use the "expert opinion scheme" in the analysis, will be listed so that the priority is to the contexts that have not been accessed yet by the users during learning.

System—[argument_602, argument_1, argument_214].

User—picks up one of the listed arguments, argument_602 as example.

System—presents the transcript of the chosen argument as shown in Figure 8.

User—starts the analysis by writing "final decision is that the death was not accident" in the root/final conclusion node, then press save.

System—divides the user statement into tokens {final, decision, death, accident}, and compares these tokens with the expert analysis for the same node, then calculates and records the error ratios for that node.

System—shows out the following message “your analysis is partially correct try to use the words {Kyle Mutch, tragic, suffered, punch}, in your node analysis, rather than the words {final, decision, ... } that have been used in the current analysis”.

User—reanalyzes the current node by adding the advised keywords.

System—compares again the current context node with the pre-existing analysis and negotiate again, guiding the user, till he/she reaches to the correct analysis for this node.

User—fills the other nodes.

System—negotiates based on the pre-existing expert analysis guiding the user during his/her analyses.

After the user finishes his/her analysis to the whole context, filling the suitable analysis for each node, the system will record the first analysis ratio for each node, then calculates and records the whole argument analysis ratio for that argument. Then the system, based on the whole analysis ratio, will advice the student either to go to the evaluation phase or return to the learning phase.

4. Related work and discussion

Early, the field of AI and education was very interesting to most of the researchers, where many instructional systems have been developed to hone students’ argumentation skills. SCHOLAR and WHY (Collins & Stevens, 1982) systems are examples for these trials. However, these systems were mainly designed to engage students in a Socratic dialog, which faces significant problems such as knowledge representations to develop a Socratic tutor (Collins & Stevens, 1982). This mainly occurred in complex domains like legal reasoning, control or preprocessing, and manipulated the natural language. Later, as a response to these difficulties, a number of argument mapping tools (Harrell, 2006; Reed & Rowe, 2003, 2004; Walton & Rowe, 2006) have been developed to foster debate among students about specific argument, using diagrams for argument representation. However, the data mining and artificial intelligence influence, which needed to guide the students to understand the relation between scientific theories and evidence and refine their argument analysis ability, are missing in these tools. Finally, Rahwan presented the ArgDf system (Rahawan & Reed, 2007; Zablith, Rahawan & Reed, 2007) through which users can create, manipulate and query arguments by using different argumentation schemes. Comparing ArgDf system to the authors’ approach, both of them sustain creating new arguments based on existing argument schemes. In addition, the ArgDf system guides the users during the creation process based on the scheme structure only, the users rely on their efforts and background to analyze the argument. However, in the authors’ approach, the users actions are monitored and guided not only by the scheme structure, but also by crucial hints devolved through the appropriate feedback. Accordingly, the analysis process is restricted by comparing the contrasting reconstruction of the users’ analysis and the pre-existing one. Such restriction helps in refining the users’ underlying classification. In the ArgDf system, searching existing arguments is revealed by specifying text in the premises or the conclusion, as well as the type of relationship between them. Then the users can choose to filter arguments based on a specific scheme. Whereas in the authors’ approach, searching the existing arguments is not only done by specifying text in the premises or the conclusion but also by providing different strategies based on different mining techniques in order to: (1) refine the learning environment by adding more flexible interoperability; (2) guarantee the retrieval of the most convenient hypotheses relevant to the subject of search; and (3) facilitate the search process by providing a different search criteria. At last, ALES enjoys a certain advantage over ArgDf system, it can trace the users progress and produce representative reports about the learners’ analysis history, which in turn excavate the proper weakness points in the learners’ analysis skills.

Student Progress Report For the Final Conclusions of Different Argument

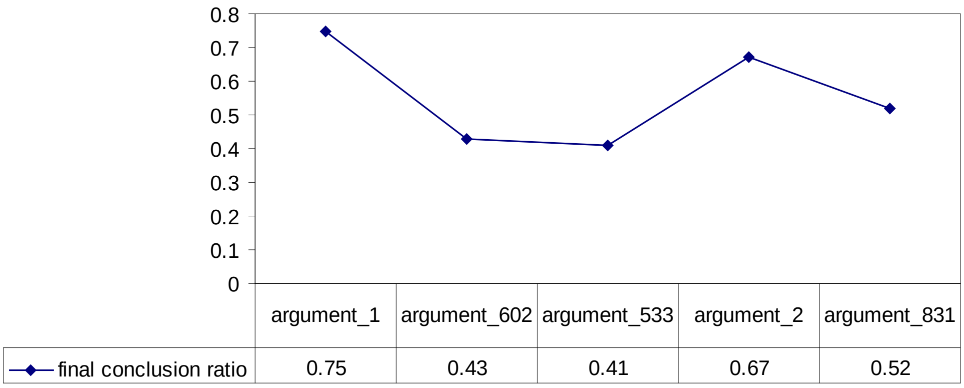


Figure 9 The resulted progress report regarding the final conclusion

Figure 9, as an example, shows the analysis progress of the current student, spotting on the conclusion node analysis ratio for different arguments using different schemes. Looking deeply in this diagram, it can be concluded that this student cannot highlight the final conclusion of different context correctly, which means that the student cannot well understand the proposed contexts.

5. Conclusion and future work

In this paper, the authors introduced an innovative argument-learning environment (ALES) to teach in argument. ALES extends the previous work done on building a highly structured argument repository with managing tools (Abbas & Sawamura, 2008b). The main aim of developing this environment is to aid in improving the students' argument skills. ALES is a model-tracing system which serves as a new trend for argument learning. The proposed architecture serves the educational process by allowing learning and assessing phases where personalized feedback is provided. ALES guides the students during argument learning, analysis and preprocessing. In addition, ALES enjoys certain advantages over others, where (1) a relevant and convenient result is assured to be obtained especially when the search statement is in this form "the Destructive War in Iraq"; (2) different representative reports that represents the students' progress can easily be extracted; and (3) two different types of personalized feedback are provided to guide the students during argument learning process. In the future, the authors intend to: (1) integrate an NLP (Neuro-Linguistic Programming) software to aid in polarity classification, in which the underlying RADB arguments are classified into affirmative and rebuttal lists to the issue of discussion; (2) use the frequent tree mining techniques (Chi & Muntz, 2001) in order to search for frequent patterns in different arguments; and (3) consider the interaction between the student model and the pedagogical model, and how this is going to affect the abductive learning phase.

References:

Abbas, S. & Sawamura, H. (2008a). Argument mining using highly structured argument repertoire. *The 1st International Conference on Educational Data Mining (EDM)*, Montreal, Qubec, Canada, 202-2010.

Abbas, S. & Sawamura, H. (2008b). A first step towards argument mining and its use in arguing agents and its usage. *The 12th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES)*, Zagreb, Croatia, 149-157.

Abbas, S. & Sawamura, H. (2008c). Towards argument mining using relational argument database. *The 2nd International Workshop* *on Juris-informatics (JURISIN),* Asahikawa Convention Bureau, Hokkaido, Japan, 22-31.

Agrawal, R. & Muntz, R. (1994). Fast algorithms for mining rules. Proceeding of the *20th VLDB Conference,* Santiago, Chile.

Aleven, V. & Ashley, K. (1997). Teaching case-based argumentation through a model and examples empirical evaluation of an intelligent learning environment. *The Eighth World Conference of the Artificial Intelligence in Education Society*, 87-94.

Baker, M., Andriessen, J. & Suthers, D. (2003). *Arguing to learn confronting cognitions in computer-supported collaborative learning environments*. Dordrecht/Boston/London: Kluwer Academic Publishers.

Chi, Y. & Muntz, R. (2001). Frequent subtree mining-an overview. *Fundamenta Informaticae*, 1001-1038.

Collins & Stevens, L. (1982). Goals and strategies of inquiry teachers. *Advances in Instructional Psychology*, 65-119.

Harrell. (2006). Using argument diagramming software in the classroom. *The 16th Biennial Workshop—Conference on Teaching Philosophy*.

Paolucc, Suthers, A. & Weiner, A. (1996). Automated advice-giving strategies for scientific inquiry. *Intelligent Tutoring Systems, Third International Conference,* Berlin, 372-381.

Rahawan, I. & Reed, C. (2007). Towards an argument interchange format. *The Knowledge Engineering Review*, 1-25.

Rahawan, I. & Sakeer, P. (2006). Representing and querying arguments on semantic web. In: Dunne, P. E. & Bench-Capon, T. J. M. (Eds.). *Computational models of argument.* IOS Press.

Reed, C. & Rowe, G. (2003). Argument research corpus. In: Huget, M. P. (Ed.). *Communication in multiagent systems, lecture notes in computer science (Vol. 2650)*. Berlin: Springer Verlag, 269-283.

Reed, C. & Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artficial Intelligence Tools, 13*, 983.

Walton, D. & Rowe, G. (2006). Araucaria as a tool for diagramming arguments in teaching and studying philosophy. *Teaching Philosophy, 29*, 111-124.

Zablith, F., Rahawan, I. & Reed, C. (2007). The foundation for a world wide argument web. *Artificial Intelligence Conference (AAAI)*, (published in the *Artificial Intelligence Journal*).

(Edited by Nicole and Lily)

(continued from Page 33)

Robert, K. J. & Merrill, S. (Eds.). (1997). *Immersion education: International perspectives.* Cambridge University Press.

Romaine, S. (1984). *The language of children and adolescents*. Oxford: Basil Blackwell.

Skutaabb, K. T. (1984). Children of guest workers and immigrants: Linguistic and educational issues. In: Edwards, J. (Ed.). *Linguistic minorities, policies and pluralism*. London: Academic Press, 17-47.

Smith, A. (1982). Nationalism, ethnic separation and the intelligentsia. In: Williams, C. (Ed.). *National separatism*. Cardiff: University of Wales Press.

Stephen, M., Richard, H. & Sarah, T. (2004). *Bilingual/immersion education: Indicators of good practice*. Ministry of Education, Wellington, New Zealand.

Waite, J. (1992). Aoteareo: Speaking for ourselves. *A discussion on the development of a New Zealand languages policy*. Wellington, New Zealand: Learning Media, Ministry of Education.

WANG, H., Claire, H. & Michael, P. (2006). *Maori student achievement on the national qualifications framework 2002-2005 fact sheet*. Ministry of Education, Wellington, New Zealand.

WANG, H. & Claire, H. (2007). *Senior secondary students' achievement at Maori-medium schools, 2004-2006.* Ministry of Education, Wellington, New Zealand.

William, F. M. (1983). U.S. language status policy and the Canadian experience. In: Cobarrubias, J. & Fishman, J. (Eds.). *Progress in language planning: International perspectives*. Berlin: Mouton de Gruyter, 173-206.

(Edited by Nicole and Lily)