International Journal of Knowledge-based and Intelligent Engineering Systems 15 (2011) 25–41
DOI 10.3233/KES-2010-0208
IOS Press

ALES: An innovative agent-based learning environment to teach argumentation

Safia Abbas[a,*] and Hajime Sawamura[b]
[a]*Faculty of Computer and Information Sciences, Ain Shams University, Abbasia, Cairo, Egypt*
[b]*Institute of Natural Science and Technology, Niigata University, 8050, 2-cho, Ikarashi, Niigata 950-2181, Japan*

Abstract. Argumentation is essential in our daily life since we argue all the time in scientific communities, parliaments, courts . . . etc. In the field of education, argumentation and argument skills reflect the students' abilities to outline a claim in a logical and convincing way and provide supportable reasons for that claim as well as identifying the often implicit assumptions that underlie the claim. This paper introduces an innovative agent-based ITS teaching environment "ALES", which concerns natural argument analysis. ALES offers two phases; learning phase and evaluation phase. The learning phase encompasses two learning strategies, learning by search and learning by assessment, in which different representative reports that follow the student progress can be produced easily. During learning by search, ALES utilizes mining techniques to expose and retrieve the underlying experts' analyses that are most relevant to the subject of search. Learning by assessment provides guidance through partial and total feedback, which guide the student analysis based on the pre-selected scheme. The evaluation phase aims to assess the student's analysis comparable to the pre-existed expert's analysis. The paper aims to (i) describe the constituent models of ALES and their functions, (ii) present the encompassed tutoring scenarios associated with an illustration of the teaching pedagogy, (iii) present a comparative study between ALES and other systems in the same field.

Keywords: Argumentation, mining techniques, intelligent learning environment

1. Introduction

Argumentation is essential in our daily life since we argue all the time in scientific communities, parliaments, courts . . . etc. In the field of education, argumentation and argument skills reflect the students' abilities to outline a claim in a logical and convincing way and provide supportable reasons for that claim as well as identifying the often implicit assumptions that underlie the claim. Recently, AI in education is interested in developing instructional systems that help students hone their argumentation skill [5]. Although argumentation skill is very important in the field of education, students' have difficulty in acquiring this skill because of their inability to follow the argument and highlight the main points of a context [10].

In response to the importance of argumentation skills in education, different mapping tools (e.g., Compendium,[1] Araucaria,[2] Rationale,[3] etc.) have been developed. These tools are designed to foster students' ability to articulate, comprehend and communicate reasoning and argumentation. The main drawbacks in these tools are the absence of an administrator to adjust the argument diagram process; In other words, guiding the students to analyze arguments based on scientific theories or evidence [15], and the lack of means of arguments' search for retrieving, classifying or summarizing arguments. Put it differently, such an idea of argument to be retrieved, summarized or classified in order to obtain useful information and make use of it from a large argument database is missing. Moreover the production of appropriate common files types that enable the user to access the same argument using different tools is missing as well. Instead, special types of

*Corresponding author. E-mail: safia@cs.ie.niigata-u.ac.jp.

[1] http://compendium.open.ac.uk/software.html.
[2] http:// araucaria.computing.dundee.ac.uk/.
[3] http://rationale.austhink.com/.

files are generated such as AML files that are primarily committed to be used by Araucaria DB only.

In this paper, in order to overcome the mentioned obstacles, we present an agent-based ITS learning environment "ALES" that invokes natural argument analysis, retrieval, and re-usage from a relational argument database (RADB). The database is considered as a highly structured argument repository managed by a classifier agent [see[1,2] for more details about the RADB]. ALES offers two phases; learning phase and evaluation phase. The learning phase encompasses two learning strategies, learning by search and learning by assessment, in which different representative reports that follow the student progress can be produced easily. During learning by search, ALES utilizes mining techniques, gathered in a classifier agent, to expose and retrieve the underlying experts' analyses that are most relevant to the students' queries. Learning by assessment provides guidance through two kinds of feedback, partial and total, which guide student analysis based on the pre-selected scheme. The evaluation phase aims to assess the student's analysis comparable to the pre-existed expert's analysis. However, the contexts which presented to the student in the evaluation phase are those which have not been accessed before during the learning phase.

The paper is firstly concerned with ALES architecture, design and implementation through which it discusses each model and the used mining techniques that are utilized by the classifier agent. Secondly, It introduces the enclosed tutoring scenarios and strategies that are used during the learning phase. Finally, the paper discusses some experimental results and provides an analytical evaluation in which different representative reports have been extracted. The paper is organized as follows. Section 2 presents the agent-based ITS learning environment architecture and its encompassed models. Section 3 illustrates the different learning modes and the used tutoring scenarios by providing an illustrative example. In Section 4, we compare our work with the most relevant work performed in the field, particularly from the motivational point of view. Finally, future work and conclusions are presented in Sections 5 and 6.

2. The architecture of ALES

The architecture of the argument learning environment(ALES) as shown in Fig. 1 consists of four main parts: (i) the domain model, (ii) the Pedagogical model comprised of three components: a parser, a classifier agent, and a teaching model, (iii) the student model, and finally (iv) the graphical user interface model "GUI" implemented using Visual C++ that provides a stable encoding application and satisfiable interface. For the purpose of this paper, the next subsections present the structure and the implementation of the domain model, the student model and the pedagogical model followed by two student-system interactive scenarios at runtime.

2.1. The domain model

The domain model is represented in the form of the relational argument database (RADB), it has been developed and implemented by us, see [2,3] for more details and discussions, which summon a huge number of arguments. These arguments were previously analyzed by experts based on Walton theory of argumentation using the AIF ontology [6,11]. The domain model can semantically be represented as a forest of a numerous directed trees [7]. 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 is permitted.

Figure 2 describes the various building blocks concerned with the RADB, using screen shots of our implemented system, such that: (a) the table "Scheme_TBL" gathers the names and the indexes for different schemes, (b) the table "Scheme_Struct_TBL" assembles the details of each scheme in "Scheme_TBL", (c) the "Data_TBL" table contains the analysis of different arguments based on different scheme structure and preserves the constraints of the AIF ontology [6] (s.t. no information node(I-node) refines another I-node).

2.2. The pedagogical model

The pedagogical model is responsible for reasoning about the student's behavior according to the student model, in order to: i) retrieve the most relevant results to the subject of search, ii) expose the corresponding argument to the selected result, iii) guide the student analysis based on the pre-existing one. The pedagogical model as seen in Fig. 1 consists of three main components: a parser, a classifier agent, and a teaching model.

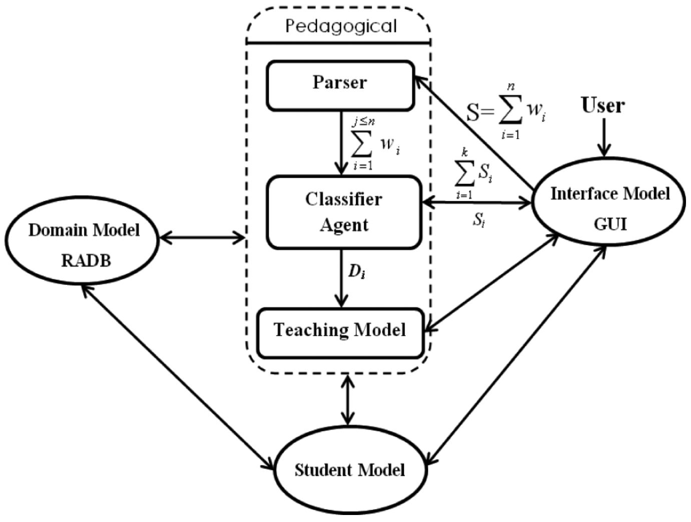
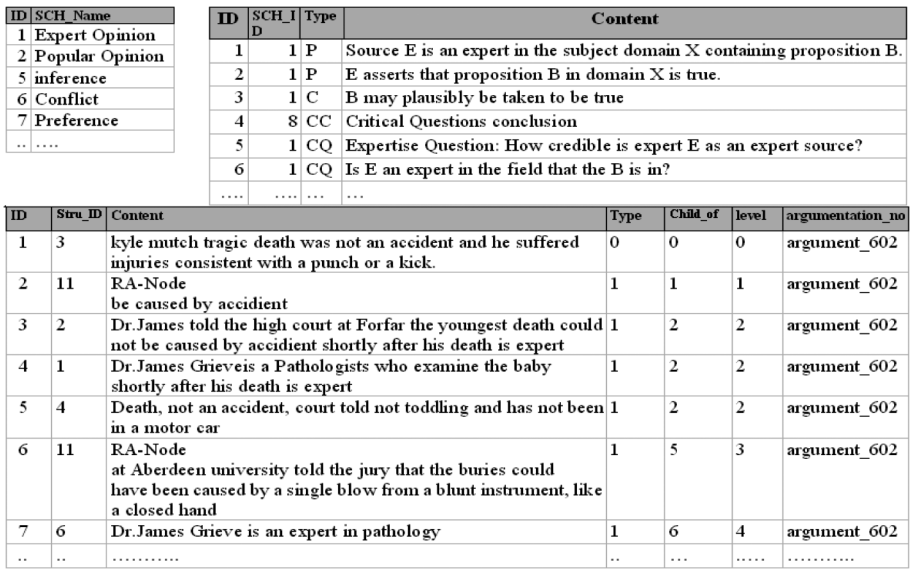


Fig. 1. ALES architecture.



ID	SCH_Name
1	Expert Opinion
2	Popular Opinion
5	inference
6	Conflict
7	Preference
..	….

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 Questions conclusion
5	1	CQ	Expertise Question: How credible is expert E as an expert source?
6	1	CQ	Is E an expert in the field that the B is in?
….	….	…	…

ID	Stru_ID	Content	Type	Child_of	level	argumentation_no
1	3	kyle mutch tragic death was not an accident and he suffered injuries consistent with a punch or a kick.	0	0	0	argument_602
2	11	RA-Node be caused by accident	1	1	1	argument_602
3	2	Dr.James told the high court at Forfar the youngest death could not be caused by accident shortly after his death is expert	1	2	2	argument_602
4	1	Dr.James Grieveis a Pathologists who examine the baby shortly after his death is expert	1	2	2	argument_602
5	4	Death, not an accident, court told not toddling and has not been in a motor car	1	2	2	argument_602
6	11	RA-Node at Aberdeen university told the jury that the buries could have been caused by a single blow from a blunt instrument, like a closed hand	1	5	3	argument_602
7	6	Dr.James Grieve is an expert in pathology	1	6	4	argument_602
..	..	………..	..	…	…..	………..

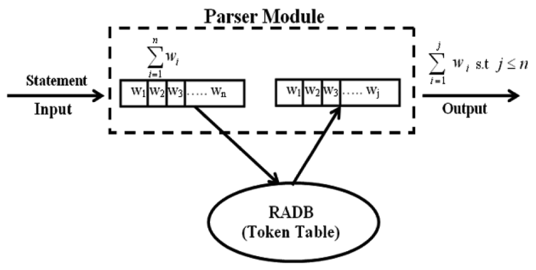
Fig. 2. The main tables in RADB.

2.2.1. *Parser*

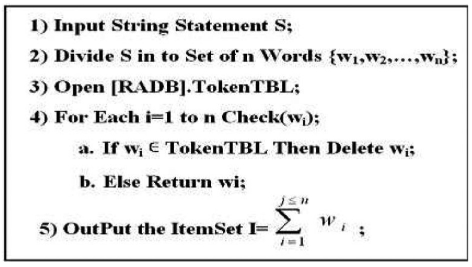
The parser, as seen in Fig. 3, receives a statement S from the student. This statement is divided by the parser into tokens, and then the number of tokens is reduced. Finally the final crucial set of words $\{w_1\ w_2 \ldots w_n\}$ is sent to the classifier agent. The tokens are reduced if they belong to a lookup table containing a set of all unnecessary words like {a, an, the, he, have, is, him …, etc.}, otherwise it is 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 i) improve the results of the classifier where combinations of unnecessary words vanish, ii) reduce the number of iterations done by the classifier agent. The parser has already been implemented and discussed in [3].

2.2.2. *Classifier agent*

The classifier agent gathers and controls different mining techniques in order to classify the retrieved contexts based on student's choices. The agent mines the RADB repository aiming to: (i) direct the search process toward hypotheses that are more relevant to student's subject of search; classifying the analogous ar-


Parser Module
$\sum_{i=1}^{n} w_i$
Statement
Input
$w_1\ w_2\ w_3 \ldots w_n$
$w_1\ w_2\ w_3 \ldots w_j$
$\sum_{i=1}^{j} w_i$ s.t $j \le n$
Output
RADB
(Token Table)

(a)



```
1) Input String Statement S;
2) Divide S in to Set of n Words {w1,w2,...,wn};
3) Open [RADB].TokenTBL;
4) For Each i=1 to n Check(wi);
      a. If wi ∈ TokenTBL Then Delete wi;
      b. Else Return wi;
5) OutPut the ItemSet I= $\sum_{i=1}^{j \le n} w_i$ ;
```

(b)

Fig. 3. The parser model.

guments in different ways based on students' choice, seeking for the most relevant arguments to the subject of search. (ii) add flexibility to the retrieving process by offering different search techniques. The agent offers three search techniques: priority search, rule extraction search, and general search. In the former, the priority search classifies and retrieves contexts based on the maximum support number and the student's search criteria using an adapted version of the AprioriTid [4] mining technique. The search criteria depends on the student's choice, it can be either by premises or by conclusion. The premises (with/against) criterion retrieves arguments by searching only in the different premises. Similarly, by conclusion retrieves the arguments by searching only in the different conclusions. The rule extraction technique summarizes the retrieved arguments searching for hidden patterns that are most relevant to the subject of search, then this 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 general search utilizes the tree structure lying in the RADB repository, and retrieves the most relevant arguments to the subject of search based on the breadth-first technique.

2.2.2.1. Priority search

The AprioriTid algorithm [4] has been implemented and embedded to the classifier agent as "Priority Search" as seen in Fig. 4. The Priority search aims to retrieve the most relevant arguments to the users' subject of search and queuing them based on the maximum support number, such that the first queued argument is the one that has more itemsets [3] 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

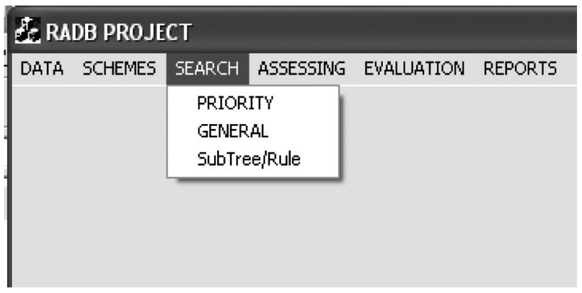
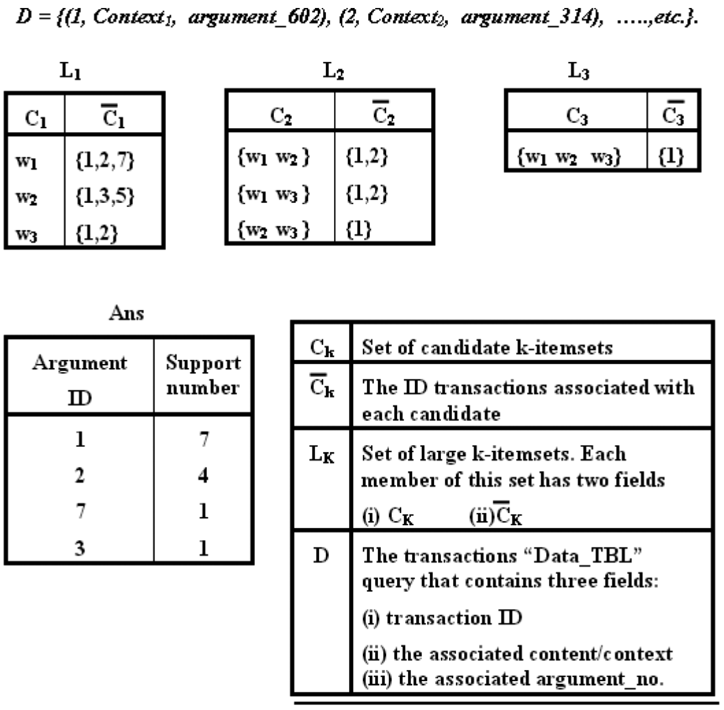


Fig. 4. The main window of our implemented system in Visual C++.

itemsets [3,4]. These combinations will then be used to classify the retrieved contexts and queued them in a descending order based on its support number. 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 Fig. 5, considers the single itemset (1-itemset) size as well as the maximum support number usage, rather than k-itemset for k$\geqslant$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 student to seek the premises, conclusions or the critical questions lying in the different arguments. For example, suppose the 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 criterion. 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. Then, the agent uses the adapted version of the AprioriTid algorithm to generate the different super itemsets $C_{2 \leqslant k \leqslant 3}$, which are the different combinations between

```
1- $L_1$={ large 1-itemsets};
2- for each itemset $C_1 \in L_1$ repeat step 5 and 6 for k=1;
3- for (k=2;$L_{k-1} \neq \Phi$; k++) do begin
4-     $C_k$ = apriori-gen($L_{k-1}$);     //new candidates
5-     for all transitions 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 $\in \cup_k \overline{C}_k$ do
9-         c.count$^{++}$;     // the support number
10- Ans={c $\in \overline{C}_k$ || descending ordered};
```

Fig. 5. An enhanced version of AprioriTid.



D = {(1, Context$_1$, argument_602), (2, Context$_2$, argument_314), ...,etc.}.

L1

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

L2

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

L3

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.

Fig. 6. The adapted AprioriTid mechanism.

different tokens. So, the generated super itemsets, as seen in Fig. 6, 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= $\{d_1, d_2, \ldots, d_m\}$ such that $\forall\ d_i \in D\ \exists\ C_{k\in\{1,2,..,j\}} \subseteq d_i$. Finally, the results will be queued in a descending order and exposed in a list, where the student can choose the argument name "Argument_314" from the list to expose the associated context and analysis.

2.2.2.2. *Rule Extraction search*

Rule extraction mining is a search technique in which argument trees are encountered to discover all hidden patterns "embedded subtrees" [7] that coincide with the relation between some objects. These objects express a set of the most significant tokens of the user's subject of search. Precisely, suppose the student wants to report some information about the relation between the "USA war" and the "weapons of mass destruction". At the beginning, the user's search statements are reduced to the most significant set of tokens by the parser [1–3]. Then, the different argument trees, pre-existing in the RADB repository, are mined in order to fetch these different tokens.

Figure 7(a) shows the analysis of an argument tree, where some enclosed nodes coincide with the student's search statements, while Fig. 7(b) shows the revealed embedded subtree. Finally, each resulted subtree is

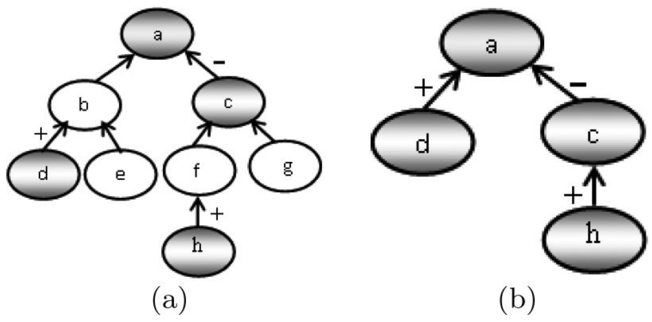
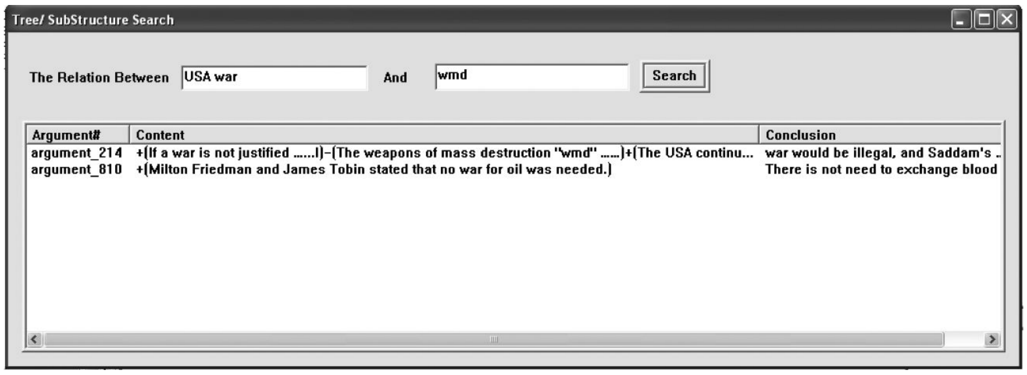


Fig. 7. The tree Rule Extraction search.



The Relation Between: USA war And: wmd

Argument#	Content	Conclusion
argument_214	+(If a war is not justified ......l)−(The weapons of mass destruction "wmd" ......)+(The USA continu...	war would be illegal, and Saddam's ..
argument_810	+(Milton Friedman and James Tobin stated that no war for oil was needed.)	There is not need to exchange blood

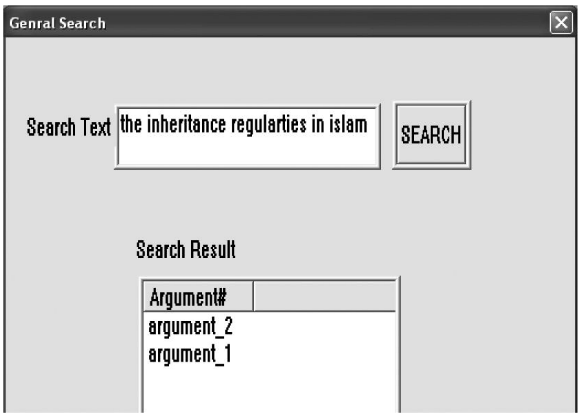
Fig. 8. The representation form of Rule Extraction search result.

expressed in the form of a rule as shown in Fig. 8, where "+" indicates that this node is a support to the final conclusion whereas "−" is a rebuttal node to the final conclusion.

2.2.2.3. General search

This technique utilizes the tree structure lying in the RADB repository, and retrieves the most relevant arguments to the subject of search. This search mechanism is considered substantial for the structured repository and annotated as "General Search" (see Fig. 9). It uses the breadth first search technique [8,14] in order to encounter all nodes in the argument trees and retrieves the most relevant group. For example, suppose the user writes "the destructive war in Iraq" as a search statement. The revealed contexts, as shown in Fig. 9, will be ordered based on the nodes' cardinality. Which means that the first queued argument is the one that contains more nodes related to the subject of search.

With respect to Fig. 10, the breadth first search seeks each tree in our RADB forest, preserving the ancestor-descendant relation [7,8] by searching first the root Fig. 10(a), then the children in the same level as in Fig. 10(b) and so on as in Fig. 10(c). Finally, if the user picks one of the resulted search arguments, the associated context and analysis are depicted.



Search Text: the inheritance regularties in islam

Search Result

Argument#	
argument_2	
argument_1	

Fig. 9. The General search representation form.

2.2.3. Teaching model

The teaching model monitors the student actions, guides the learning process and provides the appropriate feedback. The model starts its role when the classifier agent sends the document D_i selected by the student. The teaching model checks, according to the

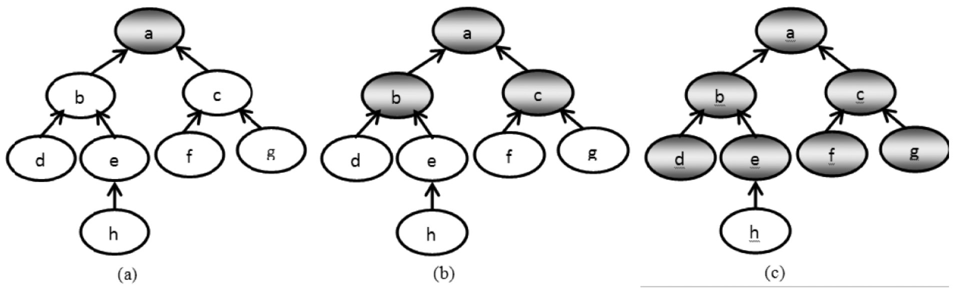


Fig. 10. The breadth first search.

current student model, whether the student is in the learning or the assessing phase. If the student is in the learning phase, the document is presented associated with the corresponding analysis. On the other hand, if the student is in the assessment phase, the student is able to do his own analysis, and the teaching model will guide him during analysis by providing personalized feedback whenever required. Two kinds of feedback are provided by the teaching model; partial argument negotiation and total argument negotiation, which are going to be discussed in Section refstudent-system interaction.

2.3. The student model

The student model stores details about student's current problem-solving state and long term knowledge progress, that is essential for future student's 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) . . ., etc. The pre-test evaluation permanently assesses the student's argument analysis skills and follows the student progress through learning process. Finally, the performance history implicitly reflects how much the student has done and how well.

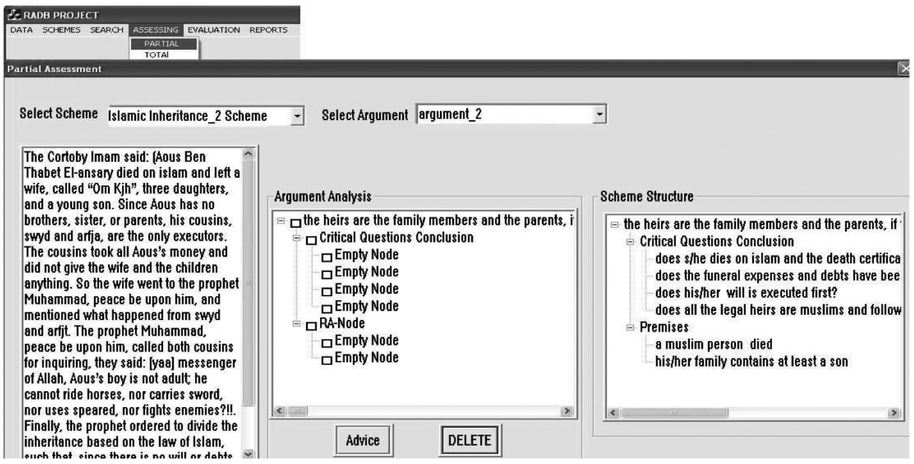
3. Tutoring scenarios

ALES offers guidance through the two tutoring phases it provides; the learning phase and the evaluation phase. The learning phase encompasses two learning strategies, learning by search and learning by assessment, in which different representative reports can be produced easily. The evaluation phase aims to assess the student's analysis comparable to the pre-existed expert's analysis then, based on the evaluation results, guide the student to either return to the learning phase or use the system as a searching tool. In both stages, the student model monitors the student's actions and stores the necessary information as part of the performance history. For example the accessed arguments, the number of access times, the assessment ratios (partial/total), and the evaluation results. Next subsections show how the student can interact with the system providing different modes. Then, an illustrative example, that shows a complete run for one of the proposed scenarios, is presented.

3.1. Different learning modes

In the learning by search mode, the teaching model offers different search types that exploit mining techniques to add flexibility to the retrieving process [see Subsection 2.2.2]. It aims to avoid student's intellectual dispersion during learning by exposing the most pertinent arguments to the subject of search. Such pertinent arguments have been analyzed and pre-stored in the RADB [2,3] by experts in the domain.

In the learning by assessment mode, ALES provides a form, as seen in Fig. 11, that allows the student to choose his required scheme for his analysis. After the scheme is selected, the system proposes a list of arguments that could be analyzed using such scheme. The student is then asked to select an argument from the proposed list. In this case, the teaching model presents the transcript of the chosen argument associated with an empty tree skeleton, as shown in Fig. 11, and asks the student to start his analysis. The student starts the analysis by copying and pasting text passages from the transcript or enters free text into the nodes. The teaching model traces each node text and divides it into a 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 two



Select Scheme: Islamic Inheritance_2 Scheme

Select Argument: argument_2

The Cortoby Imam said: (Aous Ben Thabet El-ansary died on islam and left a wife, called "Om Kjh", three daughters, and a young son. Since Aous has no brothers, sister, or parents, his cousins, swyd and arfja, are the only executors. The cousins took all Aous's money and did not give the wife and the children anything. So the wife went to the prophet Muhammad, peace be upon him, and mentioned what happened from swyd and arfjt. The prophet Muhammad, peace be upon him, called both cousins for inquiring, they said: [yaa] messenger of Allah, Aous's boy is not adult; he cannot ride horses, nor carries sword, nor uses speared, nor fights enemies?!!. Finally, the prophet ordered to divide the inheritance based on the law of Islam, such that since there is no will or debts

Argument Analysis:
- the heirs are the family members and the parents, i
  - Critical Questions Conclusion
    - Empty Node
    - Empty Node
    - Empty Node
    - Empty Node
  - RA-Node
    - Empty Node
    - Empty Node

Scheme Structure:
- the heirs are the family members and the parents, if
  - Critical Questions Conclusion
    - does s/he dies on islam and the death certifica
    - does the funeral expenses and debts have bee
    - does his/her will is executed first?
    - does all the legal heirs are muslims and follow
  - Premises
    - a muslim person died
    - his/her family contains at least a son

Fig. 11. The learning by assessment form.

kinds of feedbacks, partial or total, based on the student choice and records the student's 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 student.

- **Case of partial argument negotiation:** In this case, the student starts 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 student's current node analysis to the original one in the argument database. These hints are provided before allowing the student to proceed further in the analysis process; they aim to minimize the analysis error ratio, as much as possible, for the current analyzed node [see Subsection 3.2 and Fig. 13]. Generally, the teaching model guides with the student via the partial hints at each node till the error of the current node is minimized to a specific ratio. After then, the student is able to move to the next analysis step (i.e., node).
- **Case of total argument negotiation:** The total argument negotiation is similar to the partial argument negotiation. However, the teaching pedagogy is different in that it provides hints only at the end of the analysis process [see the Appendix and Fig. 25]. In other words, after the student builds the full analysis tree for the selected context, the system interprets and evaluates the student's analysis comparable to the pre-existing one and remarks the errors.

3.2. 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 student's actions are in bold. My illustrations to some actions will be in capital letters.

SUPPOSE THE STUDENT IN THE ASSESSING PHASE CHOOSES THE PARTIAL FEEDBACK PROPERTY. THE SYSTEM WILL GIVE THE STUDENT THE ABILITY TO SELECT SPECIFIC SCHEME TO BE USED IN HIS ANALYSIS, AS SHOWN IN Fig. 12.

User: "expert opinion scheme".

THE WHOLE ARGUMENTS, THAT USE THE "EXPERT OPINION SCHEME" IN ITS ANALYSIS, WILL BE LISTED SUCH THAT THE PRIORITY IS TO THE CONTEXTS THAT HAVE NOT BEEN ACCESSED YET BY THE USER DURING LEARNING.

System:[argument_602, argument_1, argument_214].

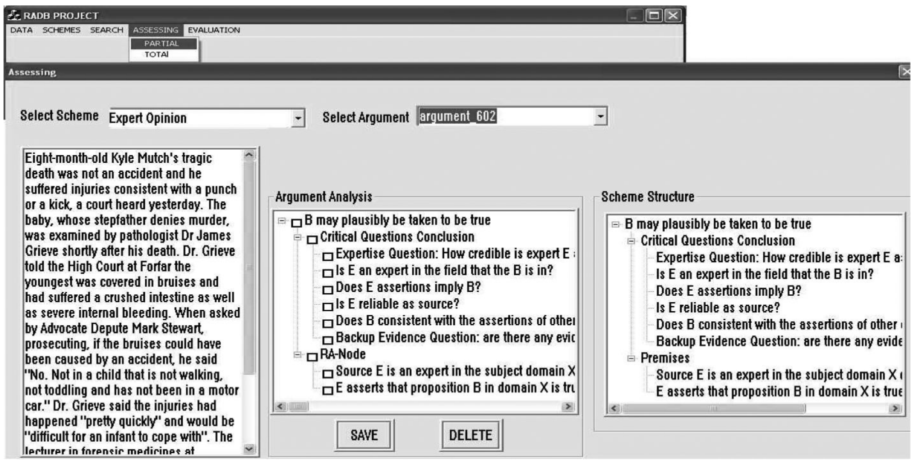
User: picks up one of the listed arguments, argument_602 as example.

System: presents the transcript of the chosen argument with an empty tree skeleton as shown in Fig. 12.

User: starts the analysis by writing "final decision is the death was not accident" in the root/final conclusion node, then the user presses 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 analy-



Select Scheme: Expert Opinion
Select Argument: argument_602

Eight-month-old Kyle Mutch's tragic death was not an accident and he suffered injuries consistent with a punch or a kick, a court heard yesterday. The baby, whose stepfather denies murder, was examined by pathologist Dr James Grieve shortly after his death. Dr. Grieve told the High Court at Forfar the youngest was covered in bruises and had suffered a crushed intestine as well as severe internal bleeding. When asked by Advocate Depute Mark Stewart, prosecuting, if the bruises could have been caused by an accident, he said "No. Not in a child that is not walking, not toddling and has not been in a motor car." Dr. Grieve said the injuries had happened "pretty quickly" and would be "difficult for an infant to cope with". The lecturer in forensic medicines at

Argument Analysis
```
□ B may plausibly be taken to be true
  □ Critical Questions Conclusion
    □ Expertise Question: How credible is expert E
    □ Is E an expert in the field that the B is in?
    □ Does E assertions imply B?
    □ Is E reliable as source?
    □ Does B consistent with the assertions of othe
    □ Backup Evidence Question: are there any evic
  □ RA-Node
    □ Source E is an expert in the subject domain X
    □ E asserts that proposition B in domain X is tru
```

Scheme Structure
```
B may plausibly be taken to be true
  Critical Questions Conclusion
    Expertise Question: How credible is expert E a
    Is E an expert in the field that the B is in?
    Does E assertions imply B?
    Is E reliable as source?
    Does B consistent with the assertions of other
    Backup Evidence Question: are there any evide
  Premises
    Source E is an expert in the subject domain X
    E asserts that proposition B in domain X is true
```

Fig. 12. The assessment form.

sis, rather than the words {final, decision,....} that have been used in the current analysis".

User: reanalyzes the current node adding the context that contains the advised keywords.

System: compares again the current context node with the pre-existing analysis and negotiate again, guiding the user as seen in Fig. 13, till he 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 analyses.

AFTER THE USER FINISHES HIS ANALYSIS TO THE WHOLE CONTEXT, FILLING THE SUITABLE ANALYSIS FOR EACH NODE, THE SYSTEM WILL RECORD THE FIRST ANALYSIS RATIO FOR EACH NODE, THEN CALCULATE AND RECORD 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.

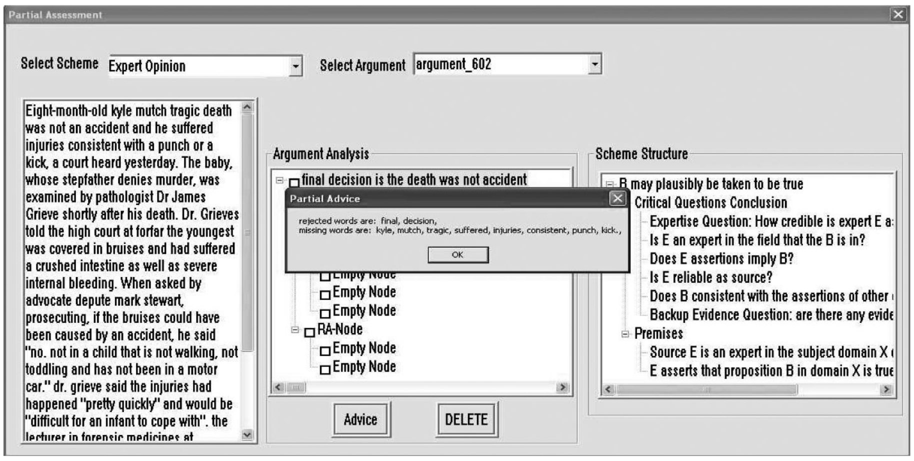
4. Discussion and experimental evaluation

Since early, the field of AI and education has been very interesting to most of the researchers, where many instructional systems have been developed to hone student's argumentation skills. SCHOLAR and WHY systems [8] are examples for these trials. However, these systems were mainly designed to engage students in a Socratic Dialog that exhibits significant problems such as knowledge representation [8], especially when used in complex domains like legal reasoning, control or preprocessing, and natural language manipulation.

Later, in response to these problems, a number of argument mapping tools [10,12,16,17] have been developed to foster debate among students about specific argument using diagrams for argument representation. However, they lack the influence of mining and artificial intelligent, which are needed to (i) guide the student to understand the relation between scientific theories and evidence through the analysis process; by providing the suitable feedback whenever required based on the student choice and the pre-existing experts analyses, (ii) retrieve the most pertinent arguments to the subject of search, which in turn avoid student's intellectual dispersion during the learning phase.

Araucaria DB [9,16], as example, is one of the most significant mapping tools that facilitates the data preprocessing and contains different analyzed contexts. However, it suffers from many problems, such as: (i) generating special types of files "AML [9]" that are primarily committed to be used by Araucaria only, (ii) users' context analysis does not follow empirical regularities and indeed analyzing any document depends on one's own thoughts and beliefs not on a dedicated structure for different argument schemes, and (iii) data retrieval methods still suffer from some problems that will be clarified in the next comparison.

To compare Araucaria DB with our system "ALES", twenty contexts were selected from Araucaria DB to act as ALES data entry. These selected contexts were reanalyzed based on Walton schemes and stored in ALES domain model "RADB", for more details about the data conversion see [1]. The contexts' content revolves around various themes including religions, crimes, Phi-


Partial Assessment

Select Scheme: Expert Opinion
Select Argument: argument_602

Eight-month-old kyle mutch tragic death was not an accident and he suffered injuries consistent with a punch or a kick, a court heard yesterday. The baby, whose stepfather denies murder, was examined by pathologist Dr James Grieve shortly after his death. Dr. Grieves told the high court at forfar the youngest was covered in bruises and had suffered a crushed intestine as well as severe internal bleeding. When asked by advocate depute mark stewart, prosecuting, if the bruises could have been caused by an accident, he said "no. not in a child that is not walking, not toddling and has not been in a motor car." dr. grieve said the injuries had happened "pretty quickly" and would be "difficult for an infant to cope with". the lecturer in forensic medicines at

Argument Analysis
- final decision is the death was not accident
  - Empty Node
  - Empty Node
  - Empty Node
  - RA-Node
    - Empty Node
    - Empty Node

Partial Advice
rejected words are: final, decision,
missing words are: kyle, mutch, tragic, suffered, injuries, consistent, punch, kick.,

Scheme Structure
- B may plausibly be taken to be true
  - Critical Questions Conclusion
    - Expertise Question: How credible is expert E a
    - Is E an expert in the field that the B is in?
    - Does E assertions imply B?
    - Is E reliable as source?
    - Does B consistent with the assertions of other
    - Backup Evidence Question: are there any evide
  - Premises
    - Source E is an expert in the subject domain X
    - E asserts that proposition B in domain X is true

Fig. 13. User guide through partial feedback.

Table 1
The retrieved results using different staments

	Araucaria DB	ALES
S_1	none	{arg_1,arg_2,arg_3}
S_2	none	{arg_1,arg_2,arg_3}
S_3	{arg_1,arg_2,arg_3}	{arg_1, arg_2,arg_3}

losophy and politics, where three of these contexts {arg_1,arg_2,arg_3} analyze the political issue for the Iraq war. In order to verify and compare the retrieval methods in both systems, three search statements related to "Iraq war" were used.

- S_1 = "the destructive war in Iraq",
- S_2 = "Iraq war",
- S_3 = "Iraq".

As shown in Table 1, the retrieved results varied in Araucaria DB rather than ALES. ALES identified the three contexts relevant to the search statements and showed them ordered by priority [see Subsection 2.2.2]. Whereas Arcauria DB could not retrieve the pertinent arguments except for S_3.

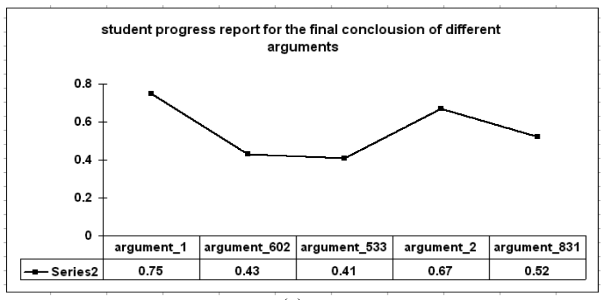
Responding to the mentioned mapping tools' problems, I. Rahwan presents the ArgDf system [6,11] through which users can create, manipulate, and query arguments using different argumentation schemes. Comparing ArgDf system to our approach, it has been found that both of them sustain creating new arguments based on existing argument schemes. The ArgDf system guides the user during the creation process based on the scheme structure only; the user relies on his efforts and his background to analyze the argument. However, in our approach, the user actions are monitored and guided not only by the scheme structure but also by crucial hints devolved through the feedback. Accordingly, the creation process is restricted by comparing the contrasting reconstruction of the user analysis and the pre-existing one. Such restriction helps in refining the user's 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 user can choose to filter arguments based on a specific scheme. Whereas in our 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 refine the learning environment by adding more flexible interoperability, guarantee the retrieval of the most convenient hypotheses relevant to the subject of search, and facilitate the search process by providing a different search criteria.

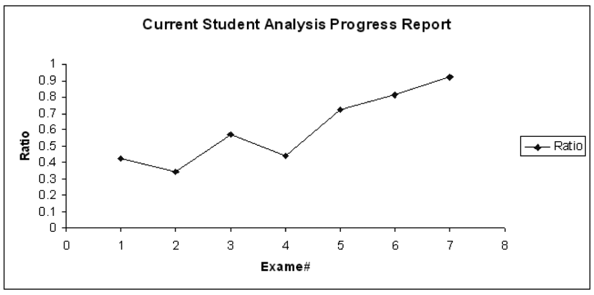
ALES enjoys certain advantages over ArgDf system and others in the same field, such as:

- it offers two tutoring strategies, learning by search and learning by assessment, through which mining techniques are used not only to retrieve convenient results but also to guide the students during the analysis process to understand the relation between the scientific theory "Walton theory" of argumentation and the evidences in the natural arguments.
- it can trace the users progress, through both learning and evaluation phases, and produce representative reports about the learner analysis history, which in turn excavate the proper weakness points in the learners' analysis skills.

Figure 14, as an example, shows the analysis progress of the current student, spotting on the conclusion node



(a)



(b)

Fig. 14. The resulted progress reports.

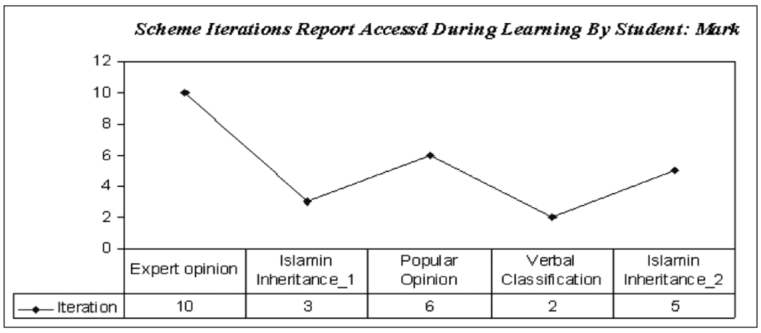


Fig. 15. The accessed schemes during learning phases and the number of access for specific student.

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 contexts correctly, which means that the student cannot well understand the proposed contexts. Whereas Fig. 14(b) shows that the student's total argument analysis skill starts to improve after the fourth exam.

Moreover, a report that declares the accessed schemes during the learning phases and the number of student's access times "iteration" can be extracted as shown in Fig. 15. The importance of this report appears in the evaluation phase, such that the scheme, which will be used in the analysis during the evaluation, either have the least number of access or have not been accessed before during the various learning phases.

On the other hand, ALES handles special types of arguments, in which only one scheme is used in the analysis process. So, if the context is much bigger and needs more than one scheme in its analysis, ALES cannot be used.

5. Future work

The work done in this paper defines argumentation as those skills that reflect the students' abilities to outline a claim in a logical and convincing way and provide supportable reasons for that claim. However, we argue that argumentation can also be defined as an intellectual process that depends on one's thoughts and beliefs and can be affected by the surrounding environment and culture. We believe that, the cultural aspects are well-reflected in the style of mutual argument analysis. Accordingly if the student's argument analyses for the different proposed arguments are traced, the student's personality, customs, culture and beliefs can be predicted. This prediction can be used in several ways to further enhance the learning process such as customizing the proposed arguments during the learning process to be related to student's culture and beliefs. It can also be used to build an intelligent student model that can analyze the data collected during the learning phase and classify the students into distinct groups confronting distinct contexts/arguments. This can be achieved through analyzing the paths the student had taken through the whole analysis processes, and the most popular schemes, or parts of schemes that had been used. In the future, we intend to use the frequent tree mining techniques [7,13] to search for the frequent patterns in different arguments in order to identify an artificial student model. In addition to integrating nat-

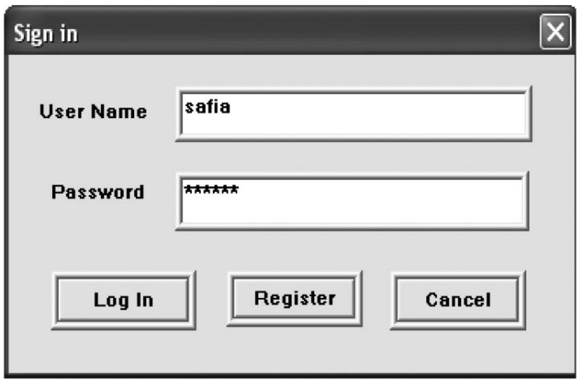


Fig. 16. The log in form.

ural language software to aid in the polarity classification, in which the underlying RADB arguments are classified into affirmative and rebuttal lists to the issue of discussion. Such classification will add more flexibility and convenience to the classifier agent.

6. Conclusions

In this paper we introduced an innovative argument learning environment (ALES) to teach argument analysis. ALES extends the previous work done on building a highly structured argument repository (RADB) with managing tool [2,3]. The main aim of developing this environment is to aid in improving the student's argument analysis skills. ALES serves as a new trend in teaching arguments. The proposed architecture serves the educational process by allowing learning and assessing phases where personalized feedback is provided. ALES guides the student during argument learning, analysis, and preprocessing. In addition, ALES enjoys certain advantages over others, where (a) a relevant and convenient result is assured to be obtained especially when the search statement is in this form: "the destructive war in Iraq", (b) special types of reports, which follow the historical learning progress, can easily be extracted.

Appendix: Practical example

This appendix shows an example of a complete run. The system interactions are written in normal font. The student's actions are in bold. My illustrations to some actions will be in capital letters.

THE SYSTEM, AS SHOWN IN Fig. 16 ASKS THE USER TO LOG IN OR REGISTER IF IT IS HIS FIRST TIME.

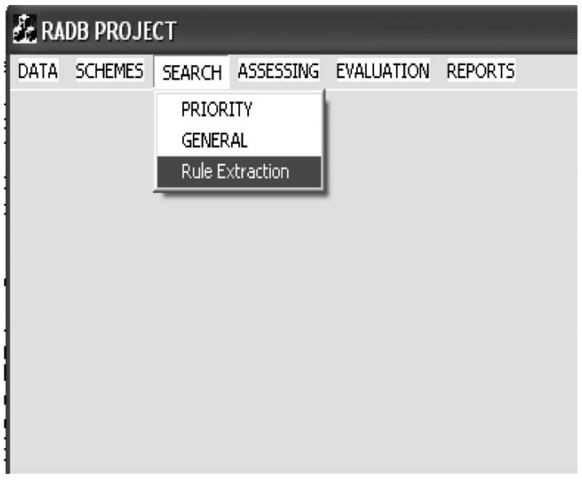


Fig. 17. The main form.

SUPPOSE THIS IS THE FIRST TIME THE STUDENT ACCESS TO THE SYSTEM.

System≫ Hello safia, do you have background about argumentation?

THE USER SHOULD CHOOSE AN ANSWER FROM AN EXISTED LIST. SUPPOSE THE USER HAVE NO IDEA ABOUT ARGUMENTATION OR ARGUMENT ANALYSIS.

User≫ No.

IN THIS CASE, THE SYSTEM GIVES THE USER A BRIEF INTRODUCTION ABOUT THE SYSTEM AND HOW HE CAN INTERACT WITH IT IN ORDER TO GAIN KNOWLEDGE ABOUT ARGUMENTATION.

System≫ Ok. Let's see how you can use the different screens. You can navigate in our system searching for specific issue using different criteria through the search tab, then from the exposed arguments list you can pick any argument. This allows you to see the original context and the associated expert analysis. Next you can use the assessing tab for quiz arguments, you are asked to analyze it based on your selected scheme. This stage can provides you with two kinds of feedback "partial or total", this feedback will help and guide you during your analysis procedure.

System≫ Are you ready to start now?
USER≫ Yes.

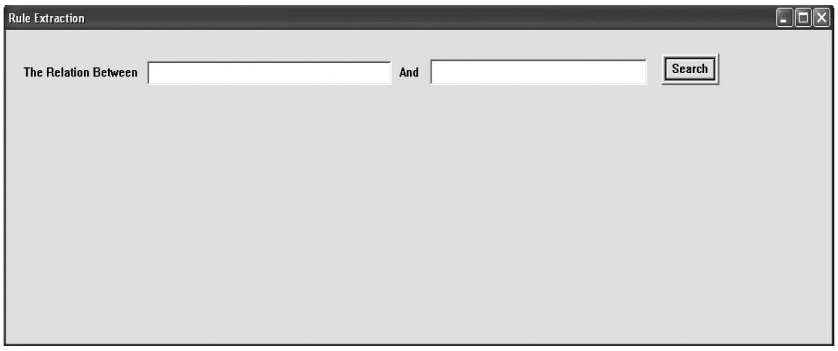
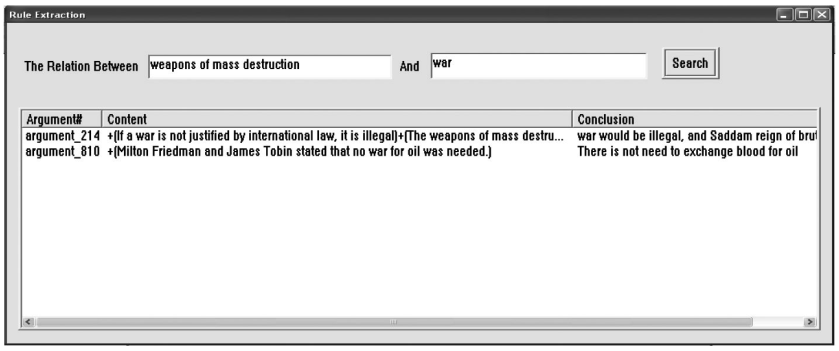


Fig. 18. The main rule extraction form.



The Relation Between: weapons of mass destruction
And: war

Argument#	Content	Conclusion
argument_214	+(If a war is not justified by international law, it is illegal)+(The weapons of mass destru...	war would be illegal, and Saddam reign of brut
argument_810	+(Milton Friedman and James Tobin stated that no war for oil was needed.)	There is not need to exchange blood for oil

Fig. 19. The rule extraction form.

THE MAIN FORM WILL APPEAR TO THE USER AS SHOWN IN 4.

User≫ presses the search tab as seen in Fig. 17 and selects the "Rule Extraction" search technique

System≫ opens the search window as seen in as seen in Fig. 18 giving the user the ability to write his search statements.

User≫ as seen in Fig. 19, the user writes in the first search text "weapons of mass destruction" and in the second text "war".
User≫ presses search button.

THE SYSTEM RETRIEVES THE DIFFERENT RELATED ARGUMENTS BASED ON RULE EXTRACTION TECHNIQUE SUCH THAT THE RESULTS, AS SEEN IN FIG. 19, WILL BE ON THE FORM OF "$+/-$" RULES RELATING TO THE FINAL CONCLUSION.

System≫ the search results are the following list [argument_214, argument_810] as in Fig. 19.

User≫ picks up argument_214 from the retrieved list to see the main context and the associated analysis.

System≫ presents the associated context and analysis as in Fig. 20.

DURING STUDENT NAVIGATION, THE STUDENT MODEL RECORDS EACH ACCESSED ARGUMENT. AFTER THE USER FINISHES NAVIGATING, HE CAN MOVE TO THE ASSESSING TAB IN ORDER TO START THE LEARNING BY ASSESSMENT PHASE, WHICH PROVIDES THE ABILITY TO ANALYZE SPECIFIC CONTEXT GUIDED BY FEEDBACK.
User≫ presses the assessing tab as seen Fig. 21 and selects the "total feedback".

Systemgg please select a specific scheme to be used in your analysis [see Fig. 22].
User≫ "expert opinion scheme".

THE WHOLE ARGUMENTS, THAT USE THE "EXPERT OPINION SCHEME" IN ITS ANALYSIS,

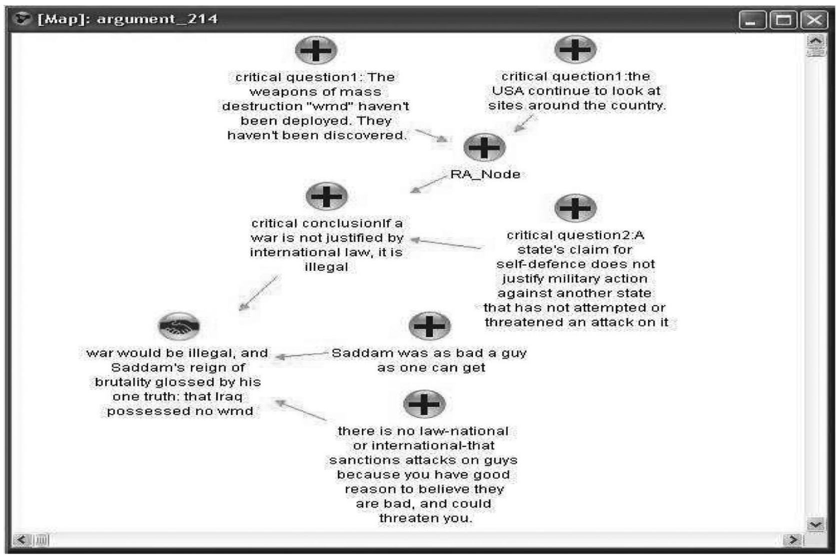

[Map]: argument_214
critical question1: The weapons of mass destruction "wmd" haven't been deployed. They haven't been discovered.
critical question1:the USA continue to look at sites around the country.
RA_Node
critical conclusionIf a war is not justified by international law, it is illegal
critical question2:A state's claim for self-defence does not justify military action against another state that has not attempted or threatened an attack on it
war would be illegal, and Saddam's reign of brutality glossed by his one truth: that Iraq possessed no wmd
Saddam was as bad a guy as one can get
there is no law-national or international-that sanctions attacks on guys because you have good reason to believe they are bad, and could threaten you.

Fig. 20. The context and analysis representation form.

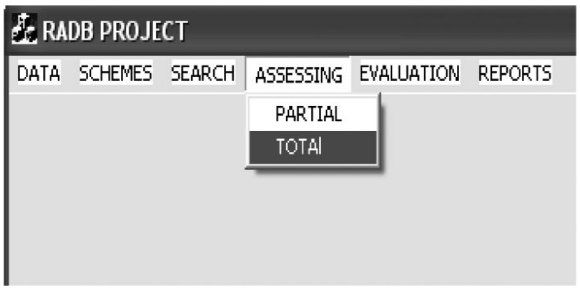


Fig. 21. The assessing tab.

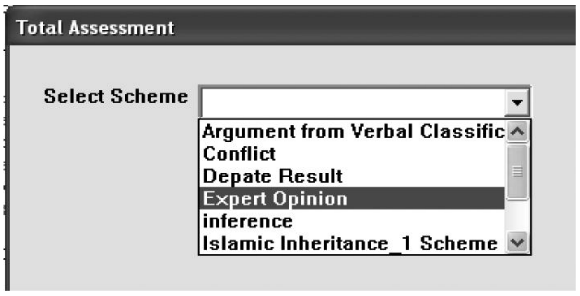


Fig. 22. The scheme selection window.

WILL BE LISTED SUCH THAT THE PRIORITY IS TO THE CONTEXTS THAT HAVE NOT BEEN ACCESSED YET BY THE USER DURING NAVIGATIONS.

System≫ [argument_533, argument_602, argument _705, . . .] as shown in Fig. 23.

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

Systemgg presents the transcript of the chosen argument associates with an empty tree skeleton, as shown in Fig. 24.

User≫ starts the analysis by copy and paste text passages from the transcript or enters free text into the nodes, as shown in Fig. 24.

Usergg presses advice button after the whole analysis is done.

Systemgg divides each statement in each node into tokens, and compares these tokens with the expert analysis for the same node.

Systemgg calculates and records the errors ratio for the whole analysis.

Systemgg shows out a declarative report, as shown in Fig. 25, that describes the mistakes of each node separately.

AS SEEN IN Fig. 25 THE STUDENT ANALYSIS OF THE FINAL CONCLUSION NODE "NODE0" IS PARTIALLY CORRECT AND THE STUDENT HAS BEEN ADVISED TO USE THESE WORDS IN HIS ANALYSIS {SADDAM, REGION,...}. ALSO IN "NODE3" WHICH IS ONE OF THE CRITICAL QUESTIONS, THE ANALYZED STATEMENT IS CORRECT HOWEVER THE TYPE OF THE NODE (SUPPORT OR ATTACH) IS WRONG. AFTER THE USER FINISHES HIS ANALYSIS TO THE WHOLE CONTEXT, FILLING THE SUITABLE ANALYSIS

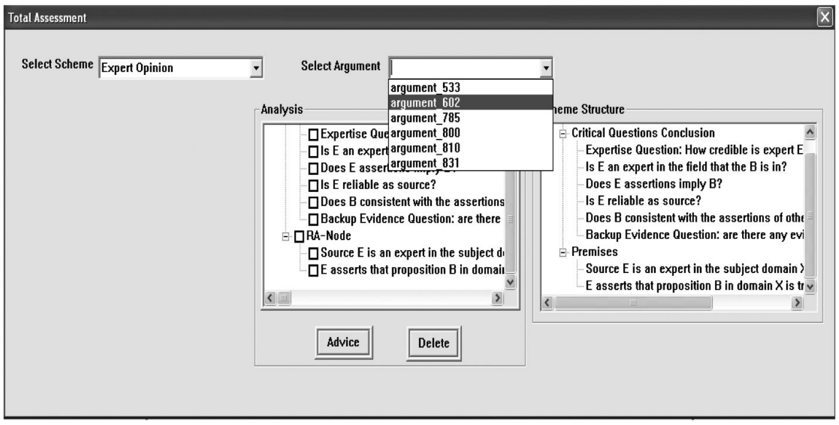
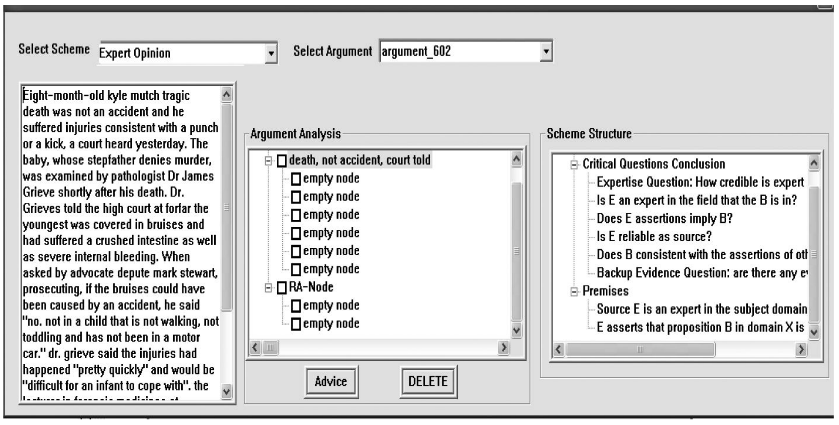

Total Assessment
Select Scheme
Expert Opinion
Select Argument
argument_533
argument_602
argument_785
argument_800
argument_810
argument_831
Analysis
Expertise Qu
Is E an expert
Does E assert
Is E reliable as source?
Does B consistent with the assertions
Backup Evidence Question: are there
RA-Node
Source E is an expert in the subject d
E asserts that proposition B in domai
Scheme Structure
Critical Questions Conclusion
Expertise Question: How credible is expert E
Is E an expert in the field that the B is in?
Does E assertions imply B?
Is E reliable as source?
Does B consistent with the assertions of othe
Backup Evidence Question: are there any evi
Premises
Source E is an expert in the subject domain X
E asserts that proposition B in domain X is tr
Advice
Delete

Fig. 23. The argument list.


Select Scheme
Expert Opinion
Select Argument
argument_602
Argument Analysis
death, not accident, court told
empty node
empty node
empty node
empty node
empty node
empty node
RA-Node
empty node
empty node
Scheme Structure
Critical Questions Conclusion
Expertise Question: How credible is expert
Is E an expert in the field that the B is in?
Does E assertions imply B?
Is E reliable as source?
Does B consistent with the assertions of ot
Backup Evidence Question: are there any e
Premises
Source E is an expert in the subject domain
E asserts that proposition B in domain X is
Advice
DELETE

Eight-month-old kyle mutch tragic death was not an accident and he suffered injuries consistent with a punch or a kick, a court heard yesterday. The baby, whose stepfather denies murder, was examined by pathologist Dr James Grieve shortly after his death. Dr. Grieves told the high court at forfar the youngest was covered in bruises and had suffered a crushed intestine as well as severe internal bleeding. When asked by advocate depute mark stewart, prosecuting, if the bruises could have been caused by an accident, he said "no. not in a child that is not walking, not toddling and has not been in a motor car." dr. grieve said the injuries had happened "pretty quickly" and would be "difficult for an infant to cope with". the

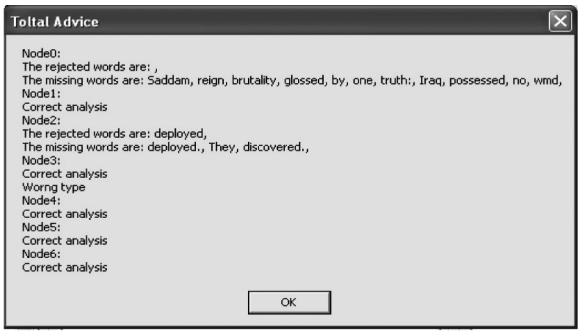
Fig. 24. The analysis window containing part of the student analysis.

FOR EACH NODE, THE SYSTEM WILL RECORD THE FIRST ANALYSIS RATIO FOR EACH NODE, THEN CALCULATE AND RECORD THE WHOLE ARGUMENT ANALYSIS RATIO FOR THAT ARGUMENT. THEN FOR EACH ASSESSMENT THE SYSTEM WILL RECORD THE CORRECTNESS RATIO TILL IT COMES TO BE MORE THAN OR EQUAL TO 90% IDENTICAL TO THE PRE-EXISTING ANALYSIS. THEN THE SYSTEM WILL ASK THE USER TO GO TO THE EVALUATION PHASE.

System≫ wow, you achieving well, it is better to accept the challenge and go to the evaluation phase.

User≫ OK, and then press the tab evaluation.

System≫ loads a context to be analyzed by the user; however the system shows a context that has not been accessed before by the user.



```
Node0:
The rejected words are: ,
The missing words are: Saddam, reign, brutality, glossed, by, one, truth:, Iraq, possessed, no, wmd,
Node1:
Correct analysis
Node2:
The rejected words are: deployed,
The missing words are: deployed., They, discovered.,
Node3:
Correct analysis
Worng type
Node4:
Correct analysis
Node5:
Correct analysis
Node6:
Correct analysis
```

Fig. 25. The resulting report.

User≫ starts to analyze without any help till press check analysis.

System≫ compares each node of the user's analysis with the expert analysis and deduces a report for the wrong nodes, and records this analysis for future progress report.

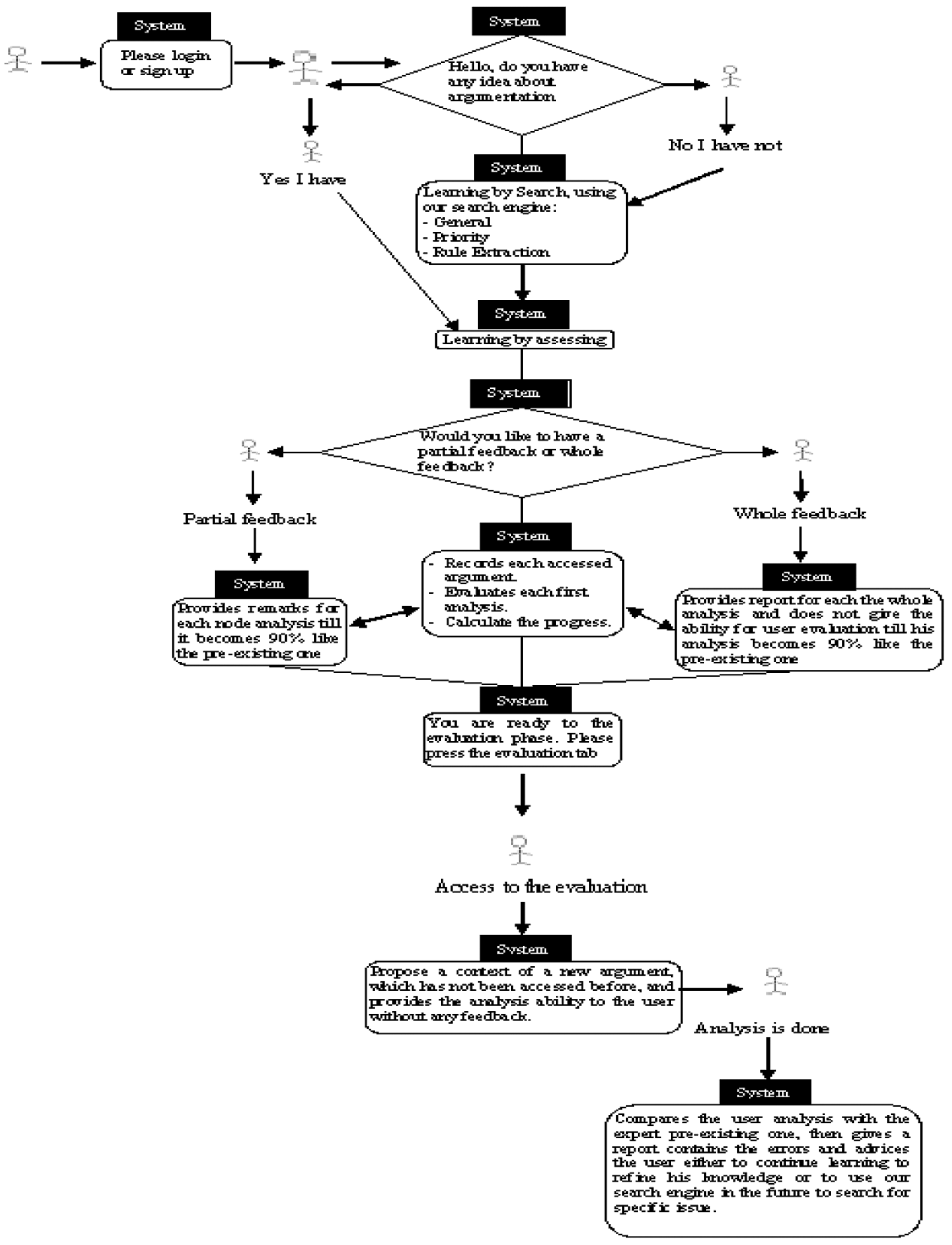

System
Please login or sign up
System
Hello, do you have any idea about argumentation
No I have not
Yes I have
System
Learning by Search, using our search engine:
- General
- Priority
- Rule Extraction
System
Learning by assessing
System
Would you like to have a partial feedback or whole feedback?
Partial feedback
Whole feedback
System
- Records each accessed argument.
- Evaluates each first analysis.
- Calculate the progress.
System
Provides remarks for each node analysis till it becomes 90% like the pre-existing one
System
Provides report for each the whole analysis and does not give the ability for user evaluation till his analysis becomes 90% like the pre-existing one
System
You are ready to the evaluation phase. Please press the evaluation tab
Access to the evaluation
System
Propose a context of a new argument, which has not been accessed before, and provides the analysis ability to the user without any feedback.
Analysis is done
System
Compares the user analysis with the expert pre-existing one, then gives a report contains the errors and adrices the user either to continue learning to refine his knowledge or to use our search engine in the future to search for specific issue.

Fig. 26. System working mechanism.

References

[1] S. Abbas and H. Sawamura, Argument mining using highly structured argument repertoire, in: *The First International Conference on Educational Data Mining* (*EDM*), *Montreal, Qubec, Canada*, 2008, pp. 202–2010.

[2] S. Abbas and H. Sawamura, A first step towards argument mining and its use in arguing agents and its, in: *12th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (*KES*), *Zagreb, Croatia*, Springer, 2008, pp. 149–157.

[3] S. Abbas and H. Sawamura, Towards argument mining using relational argument database. in: *The Second International Workshop on Juris-informatics* (*JURISIN*), *Asahikawa Convention Bureau, Hokkaido, Japan*, 2008, pp. 22–31.

[4] R. Agrawal and R. Srikant, Fast algorithms for mining rules, in: *The 20th VLDB Conference Santiago, Chile*, 1994.

[5] V. Aleven and K.D. Ashley, Teaching case-based argumentation through a model and examples empirical evaluation of an intelligent learning environment, in: *the Eighth World Conference of the Artificial Intelligence in Education Society*, 1997, pp. 87–94.

[6] I. Rahawan C. Chesnevar and C. Reed, Towards an argument interchange format. in: *The Knowledge Engineering Review*, Cambridge University Press, 2007, pp. 1–25.

[7] Y. Chi and R. Muntz, Frequent subtree mining-an overview, in: *Fundamenta Informaticae*, 2001, pp. 1001–1038.

[8] Collins and L. Stevens, Goals and strategies of inquiry teachers, in: *Advances in Instructional Psychology*, 1982, pp. 65–119.

[9] C. Reed, G. Rowe and J. katzav, Araucaria: Making up argument, in: *European Conference on Computing and Philosophy*, 2003.

[10] M. Harrell, using argument diagramming software in the classroom, in: *16th Biennial Workshop-Conference on Teaching Philosophy*, 2006.

[11] F. Zablith, I. Rahawan and C. Reed, The foundation for a world wide argument web, in: *Artificial Intelligence Conference* (*AAAI*). published in the Artificial Intelligence Journal, April 04, 2007.

[12] C. Reed, J. Katzav and G. Rowe, Argument research corpus, in: *Communication in Multiagent Systems*, (vol. 2650), M.-P. Huget, ed., Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, 2003, pp. 269–283.

[13] M.J. Zaki, Efficiently mining frequent embedded unordered trees, in: *Fundamenta Informaticae, IOS Press*, 2005, pp. 1–20.

[14] L. Looi and H. Quek, Issues in computerizing the inquiry dialogue planning process, in: *Intelligent Tutoring Systems, Third International Conference, ITS, Berlin*, 1996, pp. 252–260.

[15] A. Suthers Paolucci and A. Weiner, Automated advice-giving strategies for scientific inquiry, in: *Intelligent Tutoring Systems, Third International Conference, Berlin*, 1996, pp. 372–381.

[16] C. Reed and G. Rowe, Araucaria: Software for argument analysis, diagramming and representation, in: *International Journal on Artificial Intelligence Tools*, (vol. 13), 2004, p. 983.

[17] D. Walton and G. Rowe, Araucaria as a tool for diagramming arguments in teaching and studying philosophy, in: *Teaching Philosophy*, (Vol. 29), 2006, pp. 111–124.