Developing an Argument Learning Environment Using Agent-Based ITS (ALES)

Safia Abbas[1] and Hajime Sawamura[2]

[1] Graduate School of Science and Technology, Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 JAPAN
safia@cs.ie.niigata-u.ac.jp
[2] Institute of Natural Science and Technology,
Academic Assembly,Niigata University
8050, 2-cho, Ikarashi, Niigata, 950-2181 JAPAN
sawamura@ie.niigata-u.ac.jp

Abstract. This paper presents an agent-based educational environment to teach argument analysis (ALES). The idea is based on the Argumentation Interchange Format Ontology (AIF) using "Walton Theory". ALES uses different mining techniques to manage a highly structured arguments repertoire. This repertoire was designed, developed and implemented by us. Our aim is to extend our previous framework proposed in [3] in order to i) provide a learning environment that guides student during argument learning, ii) aid in improving the student's argument skills, iii) refine students' ability to debate and negotiate using critical thinking. The paper focuses on the environment development specifying the status of each of the constituent modules.

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 [13, 12, 16]. Recently, AI in education is interested in developing instructional systems that help students hone their argumentation skills [5]. Argumentation is classified by most researchers as demonstrating a point of view (logic argumentation), trying to persuade or convince (rhetoric and dialectic argumentation), and giving reasons (justification argumentation) [12]. Argumentation skill is extremely valuable in the educational field, and it reflects the student's abilities to outline a claim in a logical and convincing way and provides supportable reasons for the claim as well as identifying the often implicit assumptions that underlie the claim. Although argumentation skill is very important in the field of education, students' main barrier is their inability to follow the argument; highlighting the main points of a context [10]. The development of argumentation skills help students to develop their meta-cognitive and higher-order thinking abilities because argumentation requires individuals to externalize and explicitly reflect on their own thinking.

In response to the importance of argumentation skills in education, different argument mapping tools (e.g., Compendium, Araucaria, Rationale, etc.) have been developed [13]. These tools designed to foster students' ability to articulate, comprehend and communicate reasoning by producing diagrams of reasoning and argumentation. They provide a blackboard for students to record a graphical trace of their arguments. 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 [15].

In this paper, we extend our framework proposed in [3] by developing a learning environment(ALES) that uses mining agent-based ITS for teaching argument analysis. ALES uses the highly structured argument repertoire "RADB" to expose expert knowledge. It also models the student's argumentation knowledge and skills then, based on this information, it presents a group of arguments from which the user can choose one to work on. The paper is organized as follows: Section 2 introduces the learning environment (ALES) architecture. Section 3 presents an illustrative example for student-system interaction. Related work is presented in section4. Finally, conclusion and future work are illustrated in section5.

2 ALES Architecture

This section describes the architecture of the proposed learning environment(ALES) as shown in *Fig*.1. The environment consists of four main parts: the domain model represented as a highly structured argument repertoire, the Pedagogical model that contains three components: a parser, a classifier agent, and a teaching model, the student model that keeps track of the student performance and assists the pedagogical model in offering the individualized teaching, and finally the interface model "GUI". Not only does ALES teach argument analysis, but also assesses the student and guides him through personalized feedback. The next subsections illustrate the domain, student and pedagogical models in detail.

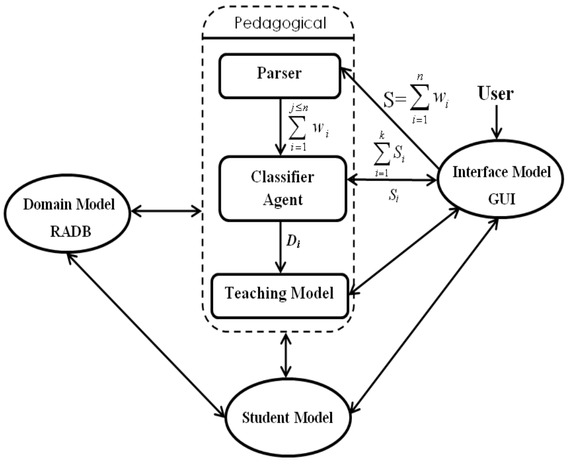


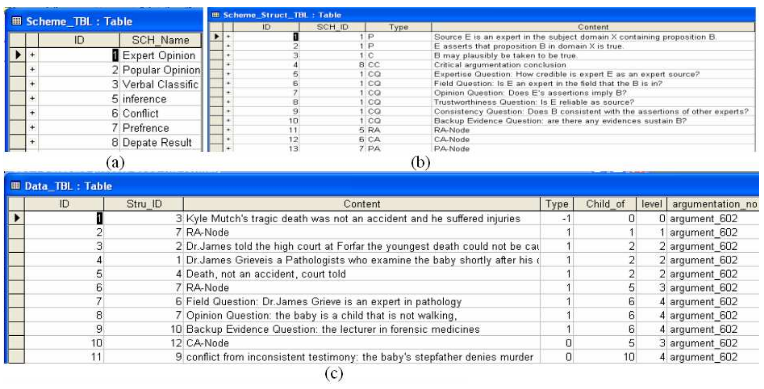
Fig. 1. ALES architecture

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 using the AIF ontology [6, 11]. The domain model can semantically be represented as a forest of a numerous directed free 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. *Fig*.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 name and the index 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). The relation between those different basic tables is shown in *Fig*.3.

2.2 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, ..., 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.



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)

Fig. 2. The main tables in RADB

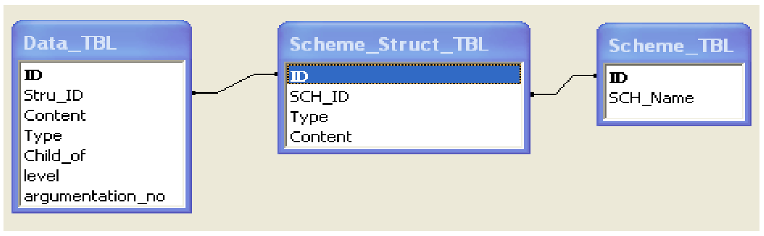


Fig. 3. The relation between the main RADB's tables

2.3 Pedagogical Model

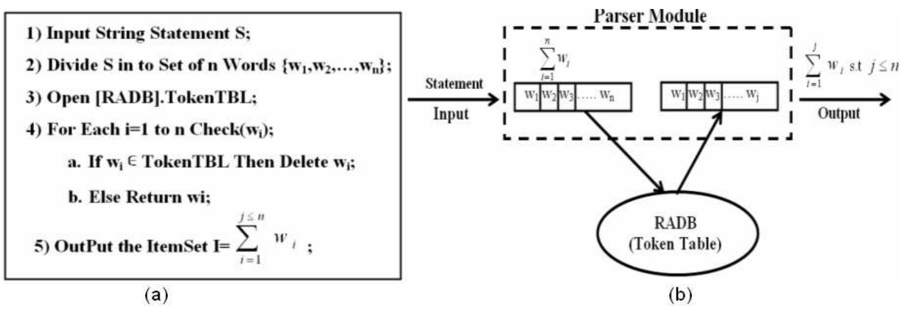
The pedagogical model is responsible for reasoning about the student behavior according to the student model, in order to: i) retrieve the most relevant results to both the subject of search and the students' background, ii) expose the corresponding argument to the selected result, iii) guide the student 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.

2.3.1 Parser

The parser as shown in *Fig*.4(b) receives a statement S from the student. This statement is divided by the parser into tokens, and then the number of tokens is reduced. 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 ..., etc }, otherwise it is added to the set of tokens to be sent to the classifier agent. Finally the final crucial set of words { w_1 w_2... w_n } is 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 is already implemented as shown in *Fig*.4(a).

2.3.2 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 towards hypotheses that are more relevant to student's subject of search; classifying the analogous arguments 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: general search, priority search, and rule extraction search. In the former, the general search classifies and retrieves the arguments based on



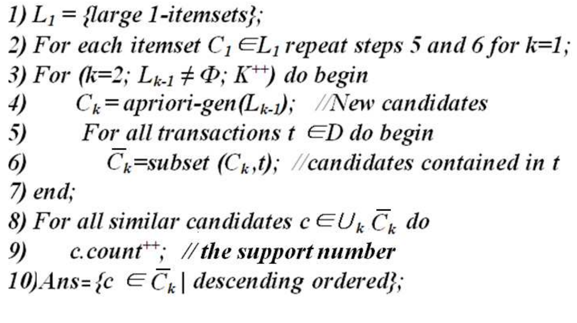
```
1) Input String Statement S;
2) Divide S in to Set of n Words {w_1,w_2,...,w_n};
3) Open [RADB].TokenTBL;
4) For Each i=1 to n Check(w_i);
      a. If w_i ∈ TokenTBL Then Delete w_i;
      b. Else Return wi;
5) OutPut the ItemSet I= $\sum_{i=1}^{j \le n} w_i$ ;
```

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

Fig. 4. The parser model

the breadth first search technique. The priority search classifies the retrieved contexts based on the maximum support number using an adapted version of the AprioriTid[4] mining technique. In the latter, the rule extraction 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.

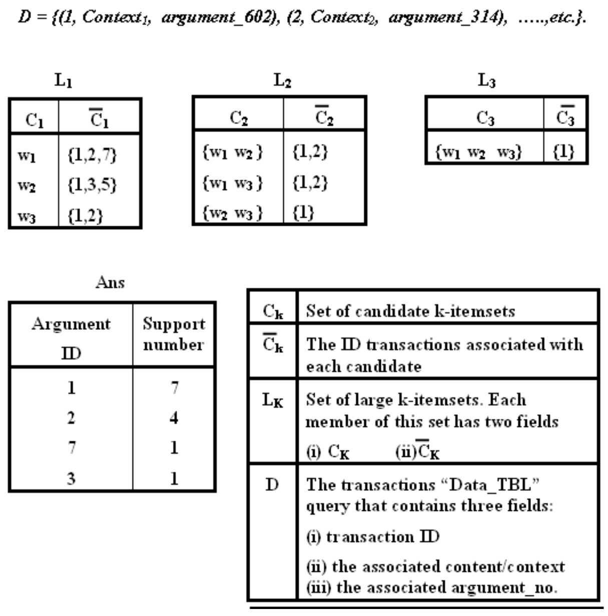
Priority Search: The AprioriTid algorithm [4] has been implemented and embedded to the classifier agent as "Priority Search". The Priority search aims to retrieve the most relevant arguments to the users' subject of search and queuing them based on the maximum support number, such that the first queued 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 itemsets [4, 3]. 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≥2 and the minimum support number "minsup" mechanism.



```
1) $L_1$ = {large 1-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-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={$c \in \overline{C}_k$ | descending ordered};
```

Fig. 5. An enhanced version of AprioriTid

For more clarification, the priority search mines specific parts of the pre-existing arguments based on the users' search criteria. This search criteria enables 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 "Iraq war". Simply, he may write "the destructive war in Iraq" as the search statement and can choose the conclusion as the search criteria.



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

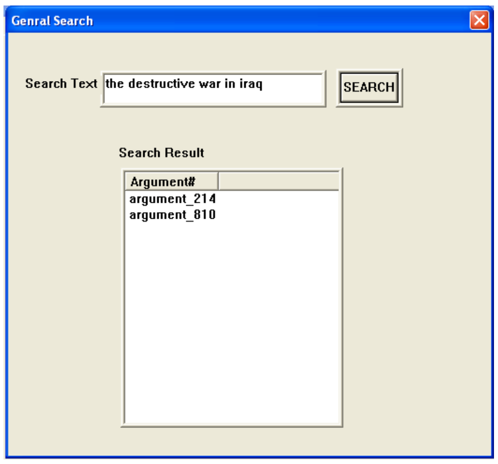
C_k	Set of candidate k-itemsets
$\overline{C_k}$	The ID transactions associated with each candidate
L_K	Set of large k-itemsets. Each member of this set has two fields (i) C_K (ii)$\overline{C_K}$
D	The transactions "Data_TBL" query that contains three fields: (i) transaction ID (ii) the associated content/context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

Fig. 6. The adapted AprioriTid mechanism

In this case, the classifier agent receives the set of significant tokens {destructive, war, iraq} 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 $C_{2\leq k\leq 3}$, which are the different combinations between 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, ..., 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 as in *Fig.* 10.

General Search: The system uses the breadth first [14] search in order to seek the different argument trees and retrieve the most relevant group. The revealed contexts are ordered based on the number of nodes "nodes cardinality" that contain any keyword, in a way where the first context is the one which has more nodes related to the search statement. For example, suppose the user writes "the destructive war in Iraq" as a search statement. The revealed contexts, as shown in *Fig.* 7, will be ordered based on the nodes' cardinality. The breadth first search seeks each tree in our RADB, preserving the ancestor-descendant relation [7] by searching first the root, then the children in the same level and so on. Finally, if the user picks one of the resulted search arguments, the associated context and analysis are depicted as shown in *Fig.* 10.

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 [2][3][1]. Then, the different argument trees, pre-existing in the RADB repository, are mined in order to fetch these different tokens. *Fig.* 8(a) shows the analysis of an argument tree, where some enclosed nodes coincide with the student's search statements, while *Fig.* 8(b) shows the revealed embedded subtree. Finally, each

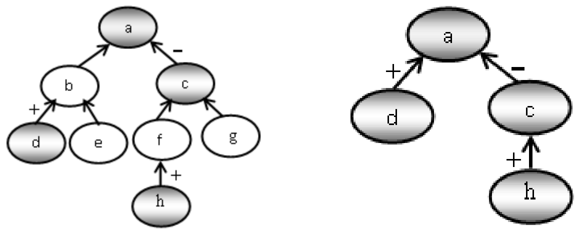


Search Text: the destructive war in iraq

Search Result

Argument#
argument_214
argument_810

Fig. 7. The General search representation form



(a)
The main argument tree

(b)
The resulted embedded subtree

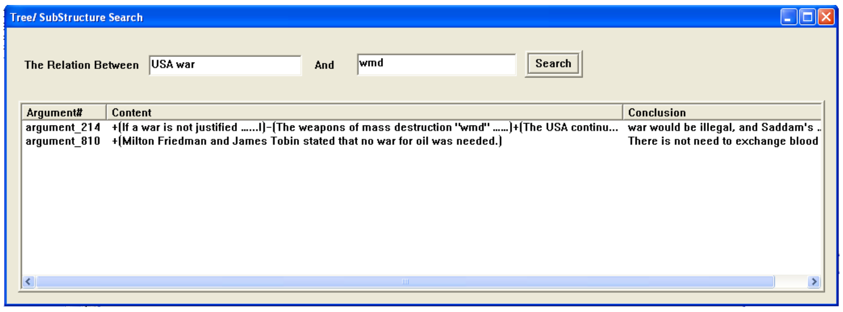
Fig. 8. The tree Rule Extraction search

resulting subtree is expressed in the form of a rule as shown in *Fig.* 9, where "+" indicates that this node is a support to the final conclusion whereas "-" is a rebuttal node to the final conclusion.

2.3.3 Teaching Model

The teaching model monitors the student actions, guides the learning process and provides the appropriate feedback. However, In the mean time, it is still in the implementation phase. 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 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 as the shown in *Fig.* 10. 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. The feedback aims to guide the student and refine his analysis and intellectual skills. Two kinds of feedback are provided by the teaching model; partial argument negotiation and total argument negotiation.

- **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. Generally, the teaching model guides with the student via the

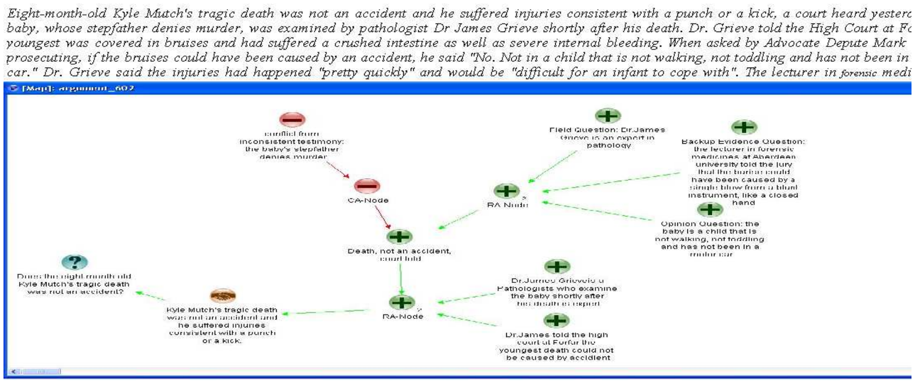


Tree/ SubStructure Search

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. 9. The representation form of Rule Extraction search result



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 yesterc
baby, whose stepfather denies murder, was examined by pathologist Dr James Grieve shortly after his death. Dr. Grieve told the High Court at Fc
youngest was covered in bruises and had suffered a crushed intestine as well as severe internal bleeding. When asked by Advocate Depute Mark
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
car." Dr. Grieve said the injuries had happened "pretty quickly" and would be "difficult for an infant to cope with". The lecturer in forensic medi

[Map]: argument_602
conflict from inconsistent testimony: the baby's stepfather denies murder
CA-Node
Field Question: Dr.James Grieve is an expert in pathology
Backup Evidence Question: the lecturer in forensic medicines at Aberdeen university told the jury that the bruises could have been caused by a single blow from a blunt instrument, like a closed hand
RA Node
Opinion Question: the baby is a child that is not walking, not toddling and has not been in a motor car
Death, not an accident, court told
Does the eight month old Kyle Mutch's tragic death was not an accident?
Kyle Mutch's tragic death was not an accident and he suffered injuries consistent with a punch or a kick.
RA-Node
Dr.James Grieve is a Pathologists who examine the baby shortly after his death is report
Dr.James told the high court at Forfar the youngest death could not be caused by accident

Fig. 10. The representation of the selected argument

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).

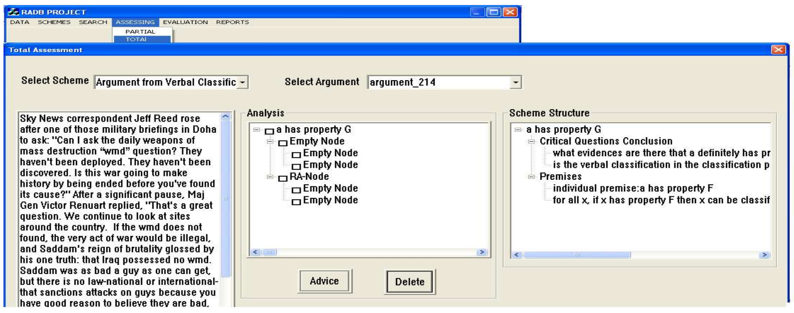
- **Case of total argument negotiation:** The total argument negotiation is similar to the partial argument negotiation. However, the teaching pedagogy is different in that it provides hints only at the end of the analysis process. In other words, after the student builds the full analysis tree for the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

Generally, in the assessing phase, the teaching model presents the transcript of the chosen argument associated with an empty tree skeleton and asks the student to start his own analysis. The student starts the analysis by copy and paste text passages from the transcript or enter free 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 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 progress of the student.

3 Illustrative Example

This example shows a complete run for the Total 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 CHOOSING THE TOTAL FEEDBACK

PROPERTY. THE SYSTEM WILL GIVE THE STUDENT THE ABILITY TO SELECT SPECIFIC SCHEME TO BE USED IN HIS ANALYSIS, AS SHOWN IN *Fig.* 11.
User≫ "Argument from Verbal Classification Scheme".



Select Scheme Argument from Verbal Classific
Select Argument argument_214

Sky News correspondent Jeff Reed rose after one of those military briefings in Doha to ask: "Can I ask the daily weapons of mass destruction "wmd" question? They haven't been deployed. They haven't been discovered. Is this war going to make history by being ended before you've found its cause?" After a significant pause, Maj Gen Victor Renuart replied, "That's a great question. We continue to look at sites around the country. If the wmd does not found, the very act of 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, but there is no law-national or international- that sanctions attacks on guys because you have good reason to believe they are bad,

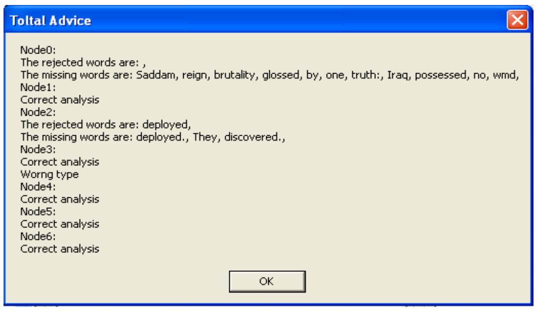
Analysis
```
a has property G
  Empty Node
    Empty Node
    Empty Node
  RA-Node
    Empty Node
    Empty Node
```

Scheme Structure
```
a has property G
  Critical Questions Conclusion
    what evidences are there that a definitely has pr
    is the verbal classification in the classification p
  Premises
    individual premise:a has property F
    for all x, if x has property F then x can be classif
```

Fig. 11. The total negotiation assessment form

THE WHOLE ARGUMENTS, THAT USE THE "ARGUMENT FROM VERBAL CLASSIFICATION 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 THE LEARNING PHASE.
System≫ [argument_214, argument_1, argument_600].
User≫ picks up one of the listed arguments, [argument_214] as example.
System≫ presents the transcript of the chosen argument with an empty tree skeleton as in *Fig.* 11.
User≫ start the analysis by copy and paste text passages from the transcript or enter free text into the nodes then press advice.
System≫ divides each statement in each node into tokens, and compares these tokens with the expert analysis for the same node. Then calculates and records the errors ratio for the whole nodes.
System≫ shows out a declarative report that describes the mistakes of each node separately.

As SEEN IN *Fig*12 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 ON OF THE CRITICAL QUESTIONS, THE ANALYZED STATEMENT IS CORRECT HOWEVER THE TYPE OF THE NODE (SUPPORT OR ATTACH) IS WRONG.



```
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. 12. The resulting report

User≫ press OK.
AFTER THE USER FINISHES HIS ANALYSIS TO THE WHOLE CONTEXT, FILLING THE

SUITABLE ANALYSIS FOR EACH NODE, THE SYSTEM WILL CALCULATES AND RECORDS THE WHOLE ARGUMENT ANALYSIS RATIO FOR THAT ARGUMENT. THIS RATIOS RECORDS WILL BE USED LATER TO EVALUATE THE PROGRESS OF THAT STUDENT.

4 Related Work

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[8] 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[8]. This mainly occurred in complex domains like legal reasoning, control or preprocessing, and manipulate the natural language. Later, as a response to these difficulties, a number of argument mapping tools[18, 10, 17, 13] 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 student to understand the relation between scientific theories and evidence, and refines his argument analysis ability, are missing in these tools.

Recently, a number of mining weblogs[9] and case-based models[5] have been proposed to tackle the mining and the artificial influence problem. The mining weblogs is considered as a classification problem for the legal or informal reasoning considering law. Though, it mines the textual data that is intractable to be processed. On the other hand, the case-based argumentation systems, such as the CATO[5], use the case based reasoning method in order to reify the argument structure through tools for analyzing, retrieving, and comparing cases in terms of factors.

Comparing CATO with our proposed application, both of them provides examples of specific issue to be studied by the different students, as well as evaluates students' arguments comparable to the pre-existing one. Regarding to the search for arguments, both systems support students' search for the existing database, and retrieve the most relevant argument. However, CATO limits the students' search by a boolean combination of factors. Also, in the full-text retrieval search, one can retrieve documents, by matching phrases, which is not relevant to the search subject. On the other hand, ALES provides different search criteria to tackle this problem, as seen in section 2, using different mining techniques in order to: summon and provide a myriad of arguments at the student's fingertips, retrieve the most relevant results to the subject of search, and organize the retrieved result such that the most relevant is the first rowed.

Finally, 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, both of them sustain creating new arguments based on existing argument schemes. In addition, 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 is not only guided by the scheme structure but also by crucial hints devolved through mining techniques. Accordingly, the creation process is restricted by comparing the contrasting reconstruction of the user's 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, facilitate the search process by providing a different search criteria.

5 Conclusion and Future Work

In this paper we introduced an agent-based learning environment (ALES) to teach argument analysis. ALES extends the previous work done on building a highly structured argument repertoire (RADB) with managing tool [2, 3]. The main aim of developing this environment is to aid in improving the student's argument 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 advantage over others, where a relevant and convenient result is assured to be obtained especially when the search statement is in this form: "the destructive war in Iraq". In the future, we intend to (i) integrate an NLP software to aid in polarity classification, in which the underlying RADB arguments are classified into affirmative and rebuttal lists to the issue of discussion, (ii) use the frequent tree mining techniques[7] in order to search for frequent patterns in different arguments, and (iii) consider the interaction between the student model and the pedagogical model, and how this is going to affect the abductive learning phase.

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*, pages 202–2010, 2008.
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*, pages 149–157. Springer, 2008.
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.*, pages 22–31, 2008.
4. R. Agrawal and R. Srikant. Fast algorithms for mining rules. In *Proceeding of 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*, pages 87–94, 1997.
6. I. Rahawan C. Chesnevar and C. Reed. Towards an argument interchange format. In *The Knowledge Engineering Review*, pages 1–25. Cambridge University Press, 2007.
7. Y. Chi and R. Muntz. Frequent subtree mining-an overview. In *Fundamenta Informaticae*, pages 1001–1038, 2001.
8. Collins and L.Stevens. Goals and strategies of inquiry teachers. In *Advances in Instructional Psychology*, pages 65–119, 1982.
9. J. Conrad and F. Schilder. Opinion mining in legal blogs. In *ICAIL07 Palo Alto, California USA*, pages 231–236, June 4-8, 2007.
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. M. Baker J. Andriessen and D. Suthers. Arguing to learn confronting cognitions in computer-supported collaborative learning environments. In *Kluwer Academic Publishers, Dordrecht/Boston/London*, volume 1, 2003.
13. C. Reed J. Katzav and G. Rowe. Argument research corpus. In *Communication in Multiagent Systems,Lecture Notes in Computer Science,Springer Verlag, Berlin, Germany*, volume 2650, pages 269–283, 2003.
14. S. Nijssen and J. N. Kok. A quickstart in frequent structure mining can make a difference. In *tenth ACM SIGKDD international conference on Knowledge discovery and data mining, USA*, 2004.
15. Suthers Paolucci, A. and A.Weiner. Automated advice-giving strategies for scientific inquiry. In *Intelligent Tutoring Systems, Third International Conference, berlin*, pages 372–381, 1996.
16. I. Rahawan and P.V. Sakeer. Representing and querying arguments on semantic web. In *Computational Models of Argument, P.E. Dunne and T.J.M. Bench-Capon (Eds.), IOS Press*, 2006.
17. C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. In *International Journal on Artificial Intelligence Tools*, volume 13, page 983, 2004.
18. D. Walton and G. Rowe. Araucaria as a tool for diagramming arguments in teaching and studying philosophy. In *Teaching Philosophy*, volume Vol. 29, pages 111–124, 2006.