Journal of Uncertain Systems
Vol.2, No.3, pp.223-240, 2008
Online at: www.jus.org.uk

Modelling Uncertainty in Refutation Selection – A POMDP Based Approach

G.S. Mahalakshmi*, T.V. Geetha
Dept. of Computer Science and Engg., Anna University, Chennai, India

Received 15 November 2007; Accepted 5 February 2008

Abstract

This paper explains the methodology behind modelling uncertainty during refutation selection in a procedural argumentation setting. The goal of argument scenario is to share valid knowledge between knowledge volunteers. The process of knowledge sharing is adapted from Indian philosophy. The exchange of arguments (or moves and counter-moves) is modeled into a Partially Observable Markov Decision Process. In this work, a new real-time belief search algorithm for selection of effective refutations and calculation of rewards is proposed and the optimal playing of arguing entities are analysed in a conversational setting.

Keywords: refutation, argument, POMDP, reward, Indian logic

1 Introduction

Argument based agent communication looks at agents not only sending messages, but support them with reasons as to why those messages are appropriate (Simon Parsons and Peter McBurney, 2003). This is a kind of persuasion where one entity tries to persuade another to adopt a belief that he or she does not currently hold (Walton and Krabbe, 1995). In such an environment, conflict situations do not only arise from misunderstandings, erroneous perceptions, partial knowledge, false beliefs, etc., but also from differences in opinions. To deal with these situations requires an argumentative capacity, the ability to handle not only demonstrative arguments but also dialectic ones. Dialogues may be viewed as dialogue games (Peter Mc Burney and Simon Parsons, 2003). As defined by Simon Parsons and Peter Mcburney (Simon Parsons and Peter McBurney, 2003), the game might start with the first player stating the reason why she believes that a proposition p is true and the other player might be bound to either accept that this reason is true (if she can find no fault with it) or to respond with the reason to why she believes it to be false. The first player is then bound by the same rules as the second was—to find a reason why the second reason is false or to accept it—and the game continues until one of the players is forced to accept the most recent reason given and thus to concede the game.

In a game theoretic framework, the combination of moves and counter-moves result in winning or losing a game. However, in a KS scenario, the objective of this argument exchange is the incremental increase in shared knowledge (Chhanda Chakraborti, 2006). Therefore, a suitable knowledge representation will result in effective argument exchange. In this work, world knowledge is represented as an ontology based on the fundamental classification of Nyaya Sastra, the famous Indian philosophy (Gradinarov, 1990; Virupakshananda, 1994; Mahalakshmi *et al.*,2002). This representation allows for effective defect exploration (Mahalakshmi and Geetha, 2007a). Moreover, Nyaya's five membered syllogistic reasoning strategies provide for effective decomposition of arguments, methodologies for defect determination and effective refutation strategy identification. Therefore, Nyaya Logics, the model for construction of world knowledge (Aghila *et al.*, 2003) based on Nyaya Sastra has been used as the basis of argument gaming paradigm (Mahalakshmi and Geetha, 2006a). In this paper, we discuss the identification of effective refutation strategies and construction of counter-arguments. The input to this identification phase is the defect set (or hole set) (Mahalakshmi and Geetha, 2007a). The effectiveness of refutation strategy is decided by the selection and validation of defects, and assignment of rewards. The attack and defeat strategies inherited from (U.Ve. Saaminatha Iyer, 1995; Virupakshananda, 1994) are adapted to suit the refutation procedures. Evaluation of arguments and

* Corresponding author. Email: mahalakshmi@cs.annauniv.edu (G.S. Mahalakshmi)

counter-arguments result in assignment of rewards, which are the indirect measure of the knowledge, expanded at the participating KS entities. To achieve this, effective inferences have to be performed for every argument exchange, which eventually results in sharing valid knowledge.

While arguing against an argument, the counter-argument(s) may include knowledge, information, explanations; preferring one counter-argument may depend on the rewards assigned for every argument exchange in reference to the discussion history in view of sharing valid information. The significance of valid knowledge may vary from entity to entity. Thus, the preference over counter-arguments cannot be determined and hence, the system is non-deterministic in deciding upon the output counter-arguments. In our assumption, argument gaming is partially observable because only partial knowledge of the state of environment is being explored by any gaming agent. The exact inferences made by the opponent from the environment are not clear. By attempting to guess at the opponent's observations, the knowledge-sharing agent attempts to guide the discussion without any deviation, in a more focused manner thus aiming to reach a definite conclusion. POMDPs are used for choosing actions when the entire world, or state space, is not always directly observable (Hoey *et al.*, 2007). By modelling the argument scenario into POMDPs, Nyaya logics can be effectively utilised to resolve knowledge inconsistencies.

2 Related Work

Partially Observable Markov Decision Processes (POMDPs) provide a very general model for sequential decision problems in partially observable environments. For POMDPs, very few researchers have explored the possibilities of online algorithms. (Geffner, 1998) used a real-time dynamic programming approach to learn a belief state estimation by successive trials in the environment. The main differences are that they do not search in the belief state tree and they need offline time to calculate their starting heuristic based on the QMDP approach. (Paquet *et al.*, 2005b) introduces RTBSS, an online POMDP algorithm useful for large, dynamic and uncertain environments. The main advantage of such a method is that it can be applied to problems with huge state spaces where other algorithms would take way too much time to find a solution. The approach is based on a look-ahead search that is applied online each time the agent has to make a decision. In this paper, we modify the algorithm RTBSS (Real-Time Belief Space Search) for real-time decision making in large POMDPs. RTBSS is particularly interesting for large real-time environments where offline solutions are not applicable because of their complexity (Paquet *et al.*, 2005a). We have used RTBSS to construct the search tree and to find the best suitable refutation. Since it is an online algorithm, it must be applied each time the agent has to make a decision. In this paper, the decision making process is interpreted as the selection of best counter-argument as response to the submitted argument. Selection of suitable counter-argument depends on the selection of appropriate and effective refutation which may attack every defect identified with the submitted argument. The following section elaborates argument gaming and the motivation of POMDP for elimination of knowledge inconsistencies during knowledge sharing.

3 Argument Gaming

The purpose of argument gaming is generally twofold: 1. to resolve contradictions, and 2. to attain a mutual consensus (M. Yamashita *et al.*, 2000). The focus is on *rational* interactions between the participating entities while involved in practical reasoning. By *rational*, we mean, giving and receiving of reasons or explanations (Katie Atkinsons *et al.*, 2005). In such a co-operative, non-monotonic argument scenario, the participating entities or gaming agents swap arguments between each other as an attempt to attack the opponent's argument. In this work, we try to tackle the interaction between argument gaming agents by assuming a gaming system for knowledge sharing which analyses the structure of defeasible arguments exchanged during the agent conversations. We have explored the essence of knowledge sharing by adapting the merging of Indian philosophy and game theoretic perspective, and hence, more detailed explanations in purview of formal agent communication (G.Vreeswijk and J. Hulstijn, 2004) is out of scope of this paper.

3.1 Nyaya Logics

Procedural argument gaming (or argument gaming) is a means of argumentative discussion between two participating knowledge sharing (KS) entities for resolving knowledge inconsistencies. In traditional times, discussion was performed as a means to eradicate one's false beliefs. The ancient school of Hinduism, Gurukula, adapted a very similar strategy of argumentative procedures for learning. According to Gotama, discussion is the adoption of one of two opposing sides (Gotama, 1930). In discussion, there is no consideration of victory or defeat (Esther Solomon, 1976). As a result correct combination of the members of syllogism and the exhaustive arguments of the opponent's objection bring about the discrimination of valid knowledge from the invalid knowledge. The Nyaya-Sutra offered a five-step inference pattern for those who want to engage in an honest, friendly, fair, and balanced debate or Vada. This five-step inference schema is known as the classical five-membered inference pattern for proper argumentation (Chhanda Chakraborti, 2006). The concern clearly was to promote the notion and the practice of a good debate, and to differentiate it from the pointless, destructive debates.

Fallacies are logical errors, which are to be avoided in any discussion. A fallacy is an object of knowledge that obstructs an inference (S.C. Vidyabhusahana, 1988). Presence of fallacies in an argument makes the argument defective (Chhanda Chakraborti, 2006). Overcoming these fallacies or defects is called “removing the holes” from the submitted argument (Virupakshananda, 1994). The syllogistic method of argument formation as cited in Nyaya philosophy (Virupakshananda, 1994) states the possibility of defects that are found identifiable in stated arguments, through which any argument can be challenged for justification and existence. (Fallacies can also be present with the manner in which the argument is proposed. Such fallacies are called Argument Fallacies (Mahalakshmi *et al.*, 2006a), which is not our scope of discussion). Refutation can be defined as pointing out the defects or fallacies in the statements of the opponent, which causes defeat to the argumentator (J.R. Ballantyne, 1849). Nyaya and Nannool state the parameterisation and classification of nature of arguments (S. Ilavarasu, 1999,U.Ve. Saaminatha Iyer, 1995, Virupakshananda, 1994). An effective refutation strategy that can cover as many defects as possible, needs to be identified and used to construct the appropriate counter-argument.

The procedural argumentation framework of Nyaya logics basically consists of collection of arguments (T.J.M. Bench-Capon, 2002). An argument is a set of propositions and can be in three states; premise (input argument), conclusion (output argument) and inference arguments that lie in the path of transition from premises to conclusion. Assignment of defeat status to an argument is generally derived from the result of various attacks, which it undergoes (Prakken and Renooji, 2001, Roos, 2000, Verheij, 2003). Every argument has a strength factor associated with it (Mahalakshmi and Geetha, 2006a; Spohn, 2002). In this paper, we assume an argument is made of concepts and relations between concepts (Mahalakshmi and Geetha, 2006a), which are modeled after the syllogistic elements of Nyaya (Oetke, 2003). Concept in the argumentation system can be: (a) subject, the current topic of discussion; (b) object of inference, whose presence / absence on the subject is to be proved (c) reason, whose relation with the object of inference either variably or invariably becomes the key feature in determining the loci or subject of existence of object of inference. Concepts are realised in the form of member qualities; quality can be mandatory, optional or exceptional and are said to describe the concept more expressively (Mahalakshmi and Geetha, 2006a). Relations explain the various means by which concept categories are related to one another. The relation between subject and object of inference is denoted by R_{S-OI}. The relation between reason and object of inference is denoted by R_{R-OI}. The relation between subject and reason is denoted by R_{S-R}. Apart from this, derived relations also tend to exist between argument concepts (Mahalakshmi and Geetha, 2006a).

An argument *A* (Mahalakshmi and Geetha, 2006a) is defined as a tuple,

$$A=< A_{id}, C_S, C_{OI}, C_R, R_{S-OI}, R_{S-R}, R_{R-OI}, A_{state}, A_{status}, A_{str}>$$

where, A_{id} - Argument index; C_S, C_{OI}, C_R - concept categories; $R_{S-OI}, R_{S-R}, R_{R-OI}$ - relation categories; A_{state} – state of argument; $A_{state} \subseteq$ {premise, inference, conclusion}; A_{status} – defeat status of arguments; $A_{status} \subseteq$ {defeated, undefeated, ambiguous, undetermined}; A_{str} – strength or conclusive force of the argument.

The defeat status of the argument is determined by evaluating the majority of valid responses among the knowledge sharing community. The evaluation of argument (also called as Reward) is deposited as the strength of the argument. The proponent proposes the argument (or counter-argument); the opponent does the evaluation. Production of effective counter-arguments lies mainly in analysing the proposed argument carefully by capturing the details of reason fallacies (or defects; (Mahalakshmi et al., 2007a)) present in the argument. Reason fallacies arise primarily due to the misinterpretation of relations existing between the concept categories: subject, reason and object of inference.

Relations fall into various types: direct, exclusive, exceptional and invariable (Mahalakshmi and Geetha, 2006a). The invariable concomitance nature of relation existence is a major factor in identifying holes of any argument and determining its defeat (Virupakshananda, 1994). Invariableness is the property, which demands the existence of relating concepts simultaneously at any instant. This invariable relation between concept entities form part of the common sense knowledge base that every KS agent is assumed to possess. Therefore, in an argument, when a concept's existence is questioned, provided one of the invariably relating concepts exists, the presence of invariableness implies existence of both the relating concepts simultaneously and hence, a separate proof for proving concept existence is rule out. Absence of invariableness projects the necessity of proof to support the concept existence. Thus, the knowledge represented in Nyaya logics provides every benefit of analysing the arguments exchanged in knowledge sharing through argument procedures. The following is the outline of partially observable non-deterministic argument games, which follow the Markov decision process in their observation of defects and projection of refutations.

3.2 The PONAG Architecture

In this work, we use the dialogue-based approach for exchanging arguments in a gaming fashion, thereby modelling the arguments in a conversational setting (Prakken, 2005). The proponent and respondent take turns by making moves. The proponent puts forward an argument designed to incur the commitment of the respondent to the conclusion. The respondent can make objections to it in the form of counter-argument. The argument and its reply have to be evaluated as a pair of moves (and counter-moves), thereby allowing new evidence to come in at a later stage of argument scenario. Fallacious arguments get weaker rewards when compared to other valid conclusive arguments (Walton, 2005). Moves (or counter-moves) of the argument process depends on identifying fallacies and constructing appropriate (and effective) refutations in opposition. A Markov decision process can be defined as a probabilistic model of a sequential decision problem, where states can be perceived exactly, and the current state and action selected determine a probability distribution on future states. A Partially Observable Markov Decision Process (POMDP) is an extension of a Markov Decision Process.

The architecture of Indian Logic-based Procedural Argumentation system, involving two knowledge-sharing (KS) agents (Simon Parsons and Peter McBurney, 2003), is shown in Figure 1. The proposed PONAG architecture (which is similar to (Irene *et al.*, 2006)) is a four-layered approach with lower two layers contributing a great deal to accomplish knowledge sharing. The knowledge base is represented in the form of Indian logic based Ontology with more specialized and enriched concepts and relations between them. Argument analysis is the preliminary stage or pre-processing stage where the input arguments are processed to identify the component elements of arguments. First layer performs the observation or hole finding mechanism out of the opponent's argument, called Defect Analyser; Second layer provides the action or defeat strategy identification, referred to as Defeat Analyser. Refuter consists of two functionalities: mapping and response generation. The response generated is in terms of counter-arguments. Mapper maps the output of the first two layers and identifies the perfect set of *defect-defeat* pairs. Responder constructs the counter-argument(s) and Evaluator evaluates the constructed counter-argument(s) as well as the inference obtained during knowledge sharing (Mahalakshmi *et al.*, 2006b). The actual output is given by the uppermost layer, Evaluator (refer Fig. 1) that executes the modified RTBSS algorithm for calculation of respective rewards.

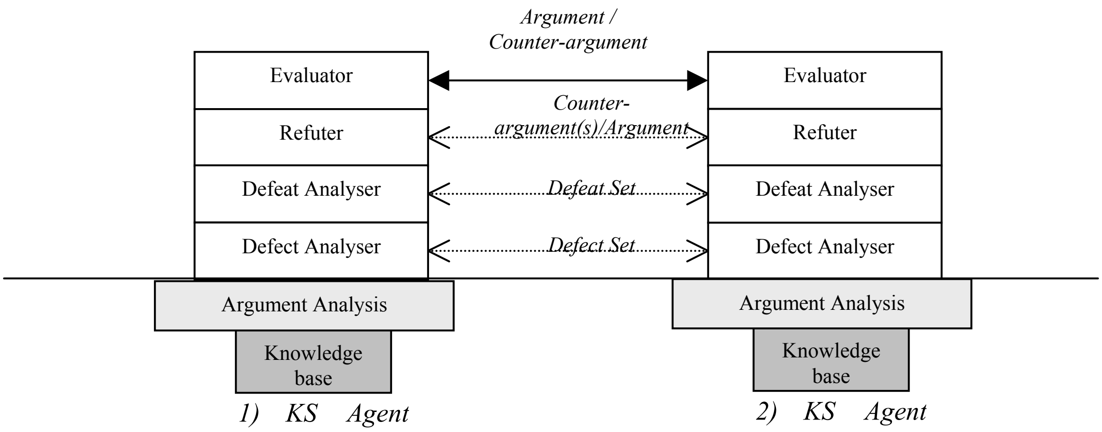

Figure 1: Partially observable non-deterministic argument gaming architecture[1] for knowledge sharing

3.3 The PONAG model

A partially observable non-deterministic argument game (PONAG) is a framework that allows imperfect information about the random arguments exchanged in any state of the system. Every agent may have different beliefs about their environment (or world) and therefore, they have difference in their respective knowledge base. This difference in knowledge base induces more (or less) defects. Therefore, counter-arguments in response to a particular argument submitted by various agents at their output also vary by default. This is the reason behind the rational arguments that are exchanged between PONAG agents while knowledge sharing (Mahalakshmi *et al.*, 2007b). Formally, we model the knowledge-sharing PONAG as a co-operative POMDP model (Paquet *et al.*, 2005a) where at each point of

[1] The PONAG architecture for Knowledge Sharing in Reference (Mahalakshmi and Geetha, 2006b) has five operational layers, with Mapper as 3rd layer and refuter as 4th layer. However, refutation involves mapping to a certain extent. Therefore, we have revised the architecture with only four operational layers thereby including Mapper as a sub-layer inside Refuter. From this section onwards, Responder is the sub-layer of Refuter; Responder is involved in counter-argument generation.

exchange of arguments, the co-operative gaming agent is in one of many possible states, but does not directly observe that state. Instead, the observations are inferred from the defects analyzed out of the opponent's submitted arguments. After receiving an observation, the gaming agent then takes an action by projecting the appropriate counter-argument that results in an immediate reward by which the status of the opponent's argument is determined. For every such gaming, the knowledge-sharing environment evolves to the next state.

Formally a PONAG model for knowledge sharing is defined as a tuple $(S, \Omega, \Delta, \Gamma, R, T, O, B)$ where: S is the set of all states; $S = \{s_1, s_2, \ldots, s_m\}$, Ω is the set of all possible observations accumulated as a defect set or hole set, $\Omega = \{h_1, h_2, \ldots, h_n\}$, Δ is the set of all possible actions or refutation strategies accumulated as defeat set, $\Delta = \{d_1, d_2, \ldots, d_p\}$, Γ is the set of all possible counter-arguments constructed over the recommended action(s) or refutation(s); $\Gamma = \{c1, c2, \ldots, cq\}$; $R(s,d,c,h,s')$ is the reward function that returns the immediate reward R_w for applying counter-argument *c* of action *d* to eliminate *h* while in state *s* that resulted in state *s'*, $T(s,d,c,s') = \Pr(s'|s,d,c)$ is the transition probability, which gives the probability of moving to state *s'* given that the gaming agent applies counter-argument *c* of action *d* from state *s*, $O(h,d,c,s') = \Pr(h|s',d,c)$ is the observation probability, which gives the probability of observing *h* in the next state *s'* after projecting counter-argument *c* of action *d*. This is also referred as anticipatory probability, since the next probable observation is anticipated, *B* is the belief vector, *b(s)* returns the probability that the gaming agent is in state *s*.

Since the currently made observations alone are not enough in deciding the nature of the current state, the gaming agent needs to take into account previous observations and actions as part of the procedural argument exchange to determine its current state, which is contained in *B*. The agent also needs to choose an action to be performed at every argument exchange. This action is determined by the policy $\pi: B \to \Delta$, which is a function that maps a belief state to the action the gaming agent should execute in this belief state. A knowledge sharing (KS) game unfolds over a finite sequence of stages of argument exchanges, which is determined by the *horizon t*, the cumulative strength of Ω of every argument exchange of the PONAG model, clearing which the discussion is assumed to have reached to a conclusion. Three important KS functions in PONAG are: observation, refutation and reward assignment.

The observation in knowledge sharing is twofold: first, is hole finding and second, defeat analysis. The gaming agent carefully analyses every argument's elements (Mahalakshmi *et al.*, 2007a) to find the holes contained, which is later assimilated into a hole set, Ω. The five types of defects (or reason fallacies) accepted by Nyaya are: 1. wandering or indecisive reasoning. 2. contradictory reasoning. 3. unestablished reasoning. 4. reasoning that requires as much proof as the thesis. 5. reasoning that is mis-timed or sublated (Gotama, 1930). A fallacy when exposed is a good reply to an opponent, whose argument is thus pointed out to be inefficient. Every combination of hole leads to recommending of system switches to the next state, tells how close the system is able to predict the opponent's flow of arguments in discussion. Partial observability is a salient feature of argument gaming because, in reality, except the abstract fundamental world knowledge that is structured as ontology, other higher-level domain knowledge tends to vary between the knowledge sharing gaming entities thereby injecting the degree of uncertainty in knowledge sharing. In this paper, we have attempted at describing the process of refutation and reward analysis.

4 Refutation in Argument Gaming

4.1 Fundamentals of Refutation

Richard Nordquist (Richard, 2008) in his Grammar & Composition defines refutation as follows: Refutation can be defined as the part of an argument where in a speaker or writer anticipates and counters opposing points of view. Refutation is the process of attacking, weakening, tearing down or destroying the argument of an opponent (David Porter, 1954). The idea is to respond to the opponent. With this, in this paper, we have used refutation to identify the appropriate and best match of hole-defeat pairs, so that, the worst hole or defect present in the input argument shall be clarified by constructing a counter-argument with the best possible and suitable defeat strategy. The prerequisite for refutation are: 1. Defect Analysis or Hole-finding mechanism 2. Defeat Analysis. In this paper, we assume the identification defects from the submitted arguments are completed by defect exploration algorithms (Mahalakshmi *et al.*, 2007a). The following section describes the process of defeat analysis.

4.2 Defeat Analysis

The defeat analyser takes the elements of arguments as input and tries to identify the best possible defeat strategy applicable over the elements of arguments. Ideas from Nyaya and Nannool (U.Ve. Saaminaatha Iyer, 1995; Virupakshananda, 1994) are adapted to suit argument defeat in our work.

4.2.1 Philosophical Recommendations of Defeat Strategies

According to Vidyabhusana (S.C. Vidyabhusana, 1988), a point of defeat or a clincher, an occasion for rebuke or a place of humiliation, arises generally from misemployment of the proposition or any other part of an argument. There are various points of defeat. They are: 1. Hurting the proposition, 2. Shifting the proposition, 3. Opposing the proposition, 4. Renouncing the proposition, 5. Shifting the reason, 6. Shifting the topic, 7. The meaningless, 8. The unintelligible, 9. The incoherent, 10. The inopportune, 11. Saying too little, 12. Saying too much, 13. Repetition, 14. Silence, 15. Ignorance, 16. Non-ingenuity, 17. Evasion, 18. Admission of an opinion, 19. Overlooking the censurable, 20. censuring the non-censurable, 21. Deviating from a tenet and 22. Semblance of a reason (S.C. Vidyabhusana, 1988).

Additional refutation strategies partially identical to Nyaya are described in Nannool. Nannool is a composition of collection of treatises, with every treatise discussing about different objectives and issues in connection with authoring books, organizing and interpreting the contents, intelligent ways of re-phrasing contents at place of duplications, teaching and learning methodology, qualities of teacher and learner and the like (S. Ilavarasu, 1999; U.Ve. Saaminatha Iyer, 1995; Vellai Vaarananaar, 1974). It is this property of knowledge sharing from Nannool, which is mapped to procedural argumentation adapted from Nyaya. Apart from various categories of style and nature of arguments as specified by Nyaya's primitive and advanced reasoning, Nannool also states further parameterization and classification of nature of arguments which when combined with Nyaya school of argument structure, frames a common grammar of handling arguments involved in knowledge sharing. The perspective of Nannool is about offline arguments which an author follows while authoring a book. However, we have modeled the perspective to suit argumentative discussion for knowledge sharing.

There are seven policies of defeat as per Nannool. They are: 1. Admission of other's opinion, 2. Opposing the proposition, 3. Accept and later refuse, 4. Introduce and support a subject, 5. Justify & conclude a subject that has equal falsivity, 6. Highlighting the defect, and 7. Evasion. Nannool states that the argument fallacies are ten in number. They are: 1. saying too little, 2. saying too much, 3. Repetition, 4. Contradiction, 5. Meaninglessness, 6. Doubtful, 7. Unintelligible, 8. Shifting the topic, 9. Renouncing the proposition and 10. Untimely. However, there are other formal and informal fallacies which are found to occur in argumentative discussions. Nannool also lists 32 techniques for argument construction. They are: 1. Causation, 2. Modularisation, 3. Summarisation, 4. Classification, 5. Conclusion by analogy, 6. Reference / Annotation, 7. Reference to conclusion, 8. Submission of other's opinion, 9. Expansion of a subject, 10. Merging, 11. Dualism, 12. Semblance of a reason, 13. Conclusion by cases, 14. Mapping to a local implication, 15. Rejecting the defeated, 16. Admission of opposition, 17. Submission of an opinion, 18. Postponing a reason, 19. Difference in conclusions, 20. Summarisation of conclusions, 21. Restricting an expansion, 22. Avoiding an expansion, 23. Partial conclusion, 24. Argument by example, 25. Justify argument by example, 26. Clearing a doubt, 27. Justify the unexplained, 28. Admission of conclusion, 29. Repeating an exception, 30. Expansion at instance, 31. Conclusion by modal, and 32. Demanding inference.

4.2.2 Categories of Defeat

Though the Indian philosophical literatures list various techniques and fallacies found in any argumentation scenario, in this paper, we have made an attempt to capture every one of the above listed ideas under five categorical argumentative discussions. The categories are classified based on the principle and objective of any argument/counter-argument exchanged in an argumentation setting. The purpose (or objective) for categorising the defeat are five in number. They are: 1. Attack, 2. Introduce, 3. Expand, 4. Change, and 5. Repeat. From this section, let us refer to the five categories by the acronym AIECR.

Attack: Attack is the operation of generating a counter-argument in opposition to the Concept: Subject or Concept: Reason. The opponent cannot oppose the object to be inferred (for known reasons) since the role of participation in the discussion makes oneself agree on the topic of discussion with the objective of proving the existence or non-existence of object of inference.

Introduce: Introduce is the idea of including a new topic which is closely related to the topic or subject of discussion, such that, the responses does not deviate very much from the topic of discussion. All concepts and relation between the three concept types can be introduced one after another; i.e. it may so happen that the current object of inference is put aside by the introduction of new object of inference (or) the conclusion of newly introduced object of inference will aid in concluding the previous object of inference and therefore, the discussion whirls around the proving or disproving of new object of inference.

Expand: Expansion is the idea of enriching or strengthening the concept or relation of the input argument.

Change: Change is the operation of generating a counter-argument which changes one or many of the elements of the input argument of the proponent. Change can happen by Introduce or Expand.
Repeat: Repeat is the idea of counter-argument generation by repeating the part or whole of the input argument.

Though attack, expand and change are the commonly recommended defeat strategies other perspectives of defeat categorisation (like, Introduce, Repeat) are also analysed to suit procedural argumentative discussion for knowledge sharing. Few defeat strategies are selected and categorised under the above five defeat perspectives (refer Fig. 2). However, a single defeat strategy shall be found to possess more than one objective of defeat which is clearly indicated in the following Figure 2.

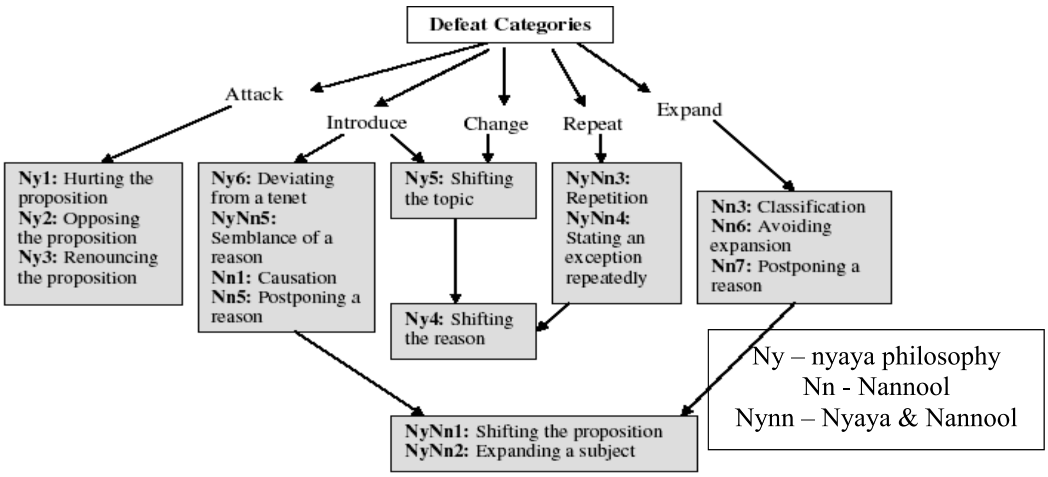

Figure 2: Defeat strategies and their classification

4.2.3 Defeat Determination by CRQ

Similar to defect analysis, the nature and type of concepts and relations of the input argument becomes the major criterion in the deciding of techniques of argument defeats. Therefore, a clear analysis of argument concepts and their relations (CRQ – Concept, Relation, Quality) across the table of defeat strategy determination (refer Table 1) will generate a defeat set for every input argument at hand. The table of defeat strategy determination is given below in Table 1. The elements E_i denote Elements of arguments. The elements are: E_1 - C_S ; E_2 - C_R ; E_3 - C_{OI} ; E_4 - $R_{S\text{-}OI}$; E_5 - $R_{S\text{-}R}$; E_6 - $R_{R\text{-}OI}$ (Mahalakshmi and Geetha, 2006a).

Construction of defeat strategy determination involves distinguishing the defeat strategies into two classes, concept-originated vs. relation originated (refer Fig. 3). Defeat strategies which can be applied to oppose (or AIECR) the concepts of the input argument are termed as concept refutations; defeat strategies that are applied to oppose the relation elements of the input argument are termed as relation refutations; there are hybrid defeat strategies which are common to both kinds. Concepts and relations refers to Indianised logics' definition of enriched concepts and relations (Mahalakshmi and Geetha, 2006a), in this paper.

4.3 Refutation Selection

The art defeat analysis has two phases: Defeat strategy determination and Defeat Selection (Mahalakshmi and Geetha, 2007c). The input argument, which is pre-processed into elements of arguments, is fed as input to defeat analysis. The elements of arguments from the input argument are matched across the table of defeat determination which generates the recommended list of defeat strategies. The list of recommended defeat strategies for a particular set of elements of arguments is termed as a defeat set. In a nutshell, defeat Analyser performs Defeat Analysis (see Figure 4), covering all possibilities of eliminating the identified holes *h*, by recommending one or more defeat strategies, d_{cat}, which may even give rise to constructing more than one counter-argument for eliminating a particular hole. The hole to be eliminated decides the selection of defeat strategy. There are numerous refutation techniques in Indian philosophy. Every defeat strategy, d_{cat}, identified may give rise to one or more defeat technique d_{type}, by which every counter-argument is constructed. The name, category, type of defeat and the elements of argument upon which the defeat is recommended is pooled into a Defeat Set, denoted as Δ.

Refutation (a) combines the hole set and defeat set, (b) identifies the best possible pair of hole(s) which can be clarified by the recommended defeat(s), or in other words, identifies the best possible defeat strategy (out of the defeat set) which covers the maximum number of holes in the hole set, and (c) generates the suitable counter-argument; The evaluator which is the final layer of argument gaming architecture takes care of analysing the generated counter-argument, assigns the reward values for concepts and relations which contributed to the generation of 'that' particular counter-argument and later, emits the counter-argument as a response to the opponent.

Table 1: Defeat strategy determination using elements of arguments

	Ny1	Ny2	Ny3	Ny4	Ny5	Ny6	NyNn 1,2	NyNn 3,4	NyNn 5	Nn1	Nn2	Nn3	Nn4	Nn5	Nn6	Nn7
Attack	E4 E5 E6	E1 E3	E2 E6													
Introduce					E2 E4 E6	XE1	E1 - E6		XE2 XE4	E2 E3 E6				!E2		
Expand							E1 E2 E4 E6					E1 E2			!E1 !E2 !E4 !E6	E1 E2 E4 E6
Change				E2 E6	E1 E4 E8											
Repeat				E1 E3				E1 E2 E4 XE1			E1 E2		E1 - E6			

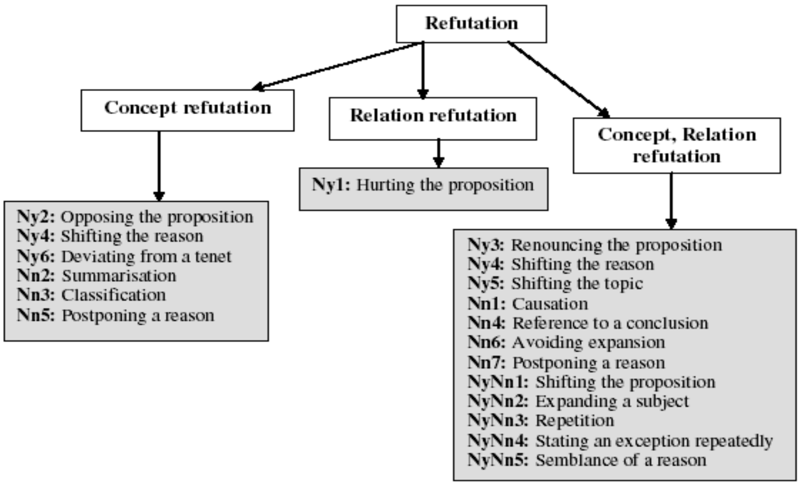

Figure 3: Classification of refutation

4.3.1 Mapper

The process of refutation is divided into two stages: Mapping and Response generation. Mapper maps the output of first two layers, Ω and and Δ, to identify every possible hole-defeat *(h-d)* pair which is represented as *h-d* matrix (see Table 2). From the matrix, the (more) perfect set of *h-d* pair(s) are selected by evaluating the elements of recommended set of *h-d* pair. The defects are indicated by h_1 to h_m. The defeats recommended are listed from d_1 to d_n. The matrix has entries marked as '1' if a particular defeat d_j is recommended for eliminating that particular hole h_i. Otherwise, the entry is marked as '0'. More than one hole may have similar defeat recommendations or in other words, a defeat technique may be recommended for elimination of more number of holes. Any evaluation to determine the nature of hole and defeat strength should consider the identical '1' entries across a row or column to select the best *h-d* pair.

The counter-argument is later constructed by the recommended defeat strategy *d,* of *h-d* pair, which is best suitable for eliminating the hole *h,* of the *h-d* pair. There may be more than one counter-argument resulting from a *h-d* pair, and the decision to select the most effective counter-argument is handled by the subsequent higher layers of knowledge sharing architecture. The objective here is not to miss the important and most productive hole, which might play a vial role in highlighting the prominent flaw in the submitted argument. Various constraints are needed to be carefully examined before deciding on the hole set (Mahalakshmi and Geetha, 2007a).With these characteristic features, the defect set Ω is mapped with defeat set, Δ, in terms of elements of arguments to identify appropriate refutation strategies.

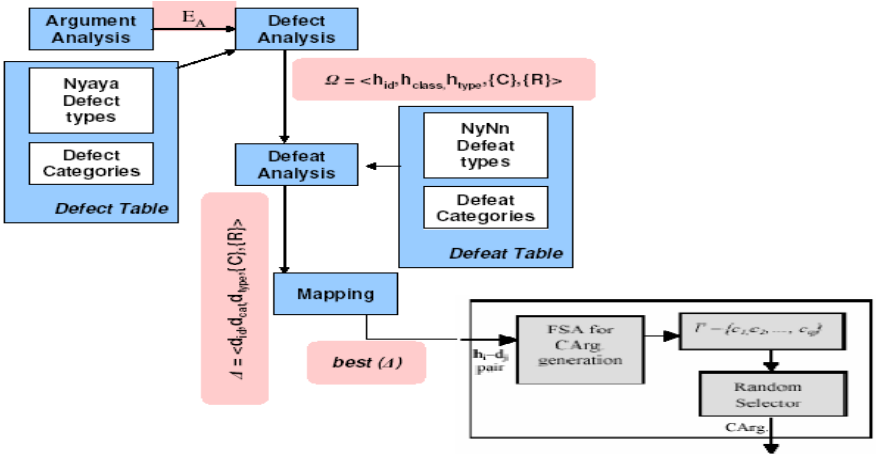

Figure 4: The process of refutation

Table 2: *h-d* matrix

defeat / hole	d_1	d_2	………………………	d_n
h_1	0	1	………………………	0
h_2	1	0	…………………………	1
…..	..	..	………………………… …..	…
h_m	0	0	…………………………	1

4.3.2 Responder

After determining hole set, Ω, and defeat set, Δ, the gaming agent is now supplied with a pool of refutations out of which the optimal refutation has to be selected. The constraint here is such that, the recommended refutation should cover the best *(h-d)* pair that will help to construct the effective counter-argument. The optimal policy $\pi_t(b)$ takes a belief state b and returns the defeat that maximizes the utility. The gaming agent simultaneously constructs all counter-arguments under the optimal refutation policy. In a scenario, where more than one counter-argument possessing equal strength factor are recommended by an optimal refutation, the counter-arguments are selected in a random manner. The projection of counter-argument receives an immediate reward and an observation. The resulting set of possible counter-arguments, $\{c_1, c_2, \ldots, c_q\}$ are populated into a refutation set, denoted as Γ. The counter-arguments in Γ are arranged with certain priorities based on certain heuristics relating to the variation of concept and relation elements of the constructed arguments. More detailed analysis on heuristics is out of scope of this paper. A random selector operates on Γ, in selecting an output counter-argument and are passed to the evaluator.

The selection of one counter-argument from the pool of generated counter-arguments is achieved after the evaluation or reward assignment for every counter-argument. Later, evaluator determines the strength of the constructed counter-argument and also the inference obtained during the hole-finding mechanism of the Defect Analyser. Determination of counter-argument strength depends on various factors: number of equivalent counter-arguments constructed, number and type of hole and defeats, number of best possible *h-d* pairs, amount of information inferred implicitly and the amount of information ignored from the submitted argument etc.

5 Theory of Reward Analysis

The objective for every KS agent is to maximize the expected sum of rewards it receives which is a direct measure of quantity and nature of holes or defects obtained for every argument exchange, through which the knowledge sharing system is expected to evolve in learning the right knowledge by discussion. PONAG is modeled after 'vada' type of debate (Virupakshananda, 1994) where the main objective is only to provide a forum for exchange of opinions and clarification of perspectives (Esther Solomon, 1976). The result of every refutation that is reflected at the argument status does not count much for calculation of rewards. The evaluation of actions made by the gaming agent along with the non-monotonic nature of knowledge obtained which is cascaded implicitly in the knowledge base, helps to improve the gaming agent's ability to behave optimally in future to achieve the valid knowledge by directing the discussion towards a definite and valid conclusion.

5.1 Modified RTBSS functions for PONAG

The PONAG demands an online policy approximation algorithm that takes as input the current belief state and returns the single action that seems to be the best for this particular belief state. In this work, the basic RTBSS algorithm (Paquet *et al.*, 2005a) has been adapted to take the specificity of dynamic decision-making in selecting the defeat strategy. The basic RTBSS (Paquet *et al.*, 2005b) calculates the action that is best suitable for any particular belief state. The real-time belief search algorithm in argument gaming for knowledge sharing proposed in this work, calculates the action c (i.e. selection of a particular counter-argument) under constrained circumstances. The constraint is the scenario where a particular defeat strategy d is applied to result in the selection of c. The defeat strategy d has to be selected such that, applying d to result in the action c should remove the maximum defects h from the submitted argument.

Formally, the value function of a belief state for a horizon of t is given by:

$$V_t(b) = R(b,d,c,h) + \gamma \max_{d\in\Delta} \sum_{h\in\Omega} P(h \mid b,d,c) V_{t-1}(\tau(b,d,c,h)) \,. \tag{1}$$

$$R(b,d,c,h) = \sum_{s\in S} b(S)R(S) \,. \tag{2}$$

$R(b,d,c,h)$ is the expected reward for the belief state. The observation probability O is given by

$$P(h \mid b,d,c) = \sum_{s'\in S} O(s',h,d,c) \sum_{s\in S} T(s,d,c,s')b(s) \,. \tag{3}$$

Also, $\tau(b,d,c,h)$ is the belief update function. It returns the resulting belief state if action d is done in belief state b and observation h is perceived. If $b' = \tau(b,d,c,h)$, then,

$$b'(s') = \eta O(s',h,d,c) \sum_{s\in S} T(s,d,c,s')b(s) \tag{4}$$

where η is the normalizing constant. Finally the policy can be obtained as:

$$\pi_t(b) = \arg\max_{d\in\Delta} \left[R(b,d,c,h) + \gamma \sum_{h\in\Omega} P(h \mid b,d,c) V_{t-1}(\tau(b,d,c,h)) \right] . \tag{5}$$

5.2 Calculation of Reward (Rw)

According to section 3.5, R_w is the immediate reward given to that entity for applying counter-argument c to eliminate hole h in the submitted argument a. This is a measure of quantity and nature of holes or defects obtained for every argument exchange. Let us assume, the knowledge sharing environment is made up of N knowledge volunteers. For every domain, let us assume, there are m no. of volunteers who are members of that domain. Let n represent the quantity of non-members of that particular domain (i.e. $N = m + n$). A member of any domain is represented by D_m and a non- member of any domain is represented by D_n. Let the variable $AttemptD_m$ represent the no. of members who attempt to answer the submitted argument. Let the variable $AttemptD_n$ represent the no. of non-members who attempt to answer the submitted argument.

Let i represent valid counter-argument and j represent invalid (or not so significant) counter-argument. Then, D_{mi} indicate the total no. of domain members giving valid counter-arguments to the submitted argument; D_{ni} indicate the total no. of non-domain members giving valid counter-arguments to the submitted argument; D_{mj} indicate the total no. of domain members giving invalid counter-arguments to the submitted argument; D_{nj} indicate the total no. of non-domain members giving invalid counter-arguments to the submitted argument.

Let tot be the total no. of (valid and invalid) counter-arguments for the submitted argument. Let l be the total no. of different counter-arguments for the submitted argument. If there are more than one valid counter-arguments, in this work, we assume the majority of valid counter-arguments be taken as the valid response for the submitted argument and the rest shall be considered as invalid counter-arguments. From this point forward, we shall replace the word '*majority*' for valid and '*other*' for invalid, when we deal with counter-arguments. Let *M* be the total no. of *majority* counter-arguments, i.e. $M \subseteq tot$ then, the probability of getting a *majority* counter-argument is given by:

$$p(majority) = \frac{\left(\frac{D_{mi}}{M} + \frac{D_{ni}}{M}\right)}{tot}. \tag{6}$$

The probability of getting *other* counter-argument is given by:

$$p(other) = \frac{\left(\frac{D_{mj}}{tot - M} + \frac{D_{nj}}{tot - M}\right)}{tot}. \tag{7}$$

The counter-argument of a knowledge volunteers is found to be either valid / invalid and unique / similar. Every counter-argument which flows through the discussion tend to fit in any one of the above-said categories with respect to the answering process. Therefore, the probability of a counter-argument of a volunteer being a *majority* counter-argument is given by

$$p(majority/other) = \frac{p(majority) * p(other/majority)}{p(other)}. \tag{8}$$

Let $K\uparrow$ denote the new knowledge gained (or the knowledge increment) during argument exchange. The reward rew is then calculated as

$$rew = AttemptD_m + AttemptD_n + M + \sum_{k=1}^{l-1} p\left(\frac{majority}{other}\right) + K_{\uparrow}. \tag{9}$$

The index '*k*' iterates over *l-1* counter-arguments, since one KS entity is supposed to have posted the argument over the discussion forum. The increment in knowledgebase (or knowledge gain), measured by $K\uparrow$ is calculated as:

$$K_{\uparrow} = \sum_{k=0}^{w-1} IC(E_A[k]) \tag{10}$$

where, IC is the information content of E_A, the elements of arguments (Mahalakshmi and Geetha, 2006a) harvested from the submitted argument and *w* being the no. of (concept and relation) words present in the argument. i.e. a word may be either a concept or a relation, when it is expressed as part of the argument. As described earlier, the elements of arguments may be composed of concepts and/or relations; E_A is a pair (C,R) where *C* and *R* denote concept and relation elements of Nyaya Logics (Mahalakshmi and Geetha, 2007a). The cumulative sum of concepts is denoted by C_{pr}, (pr – priority factor; which prioritises the element within the argument) and the cumulative sum of relations is denoted by R_{pr} (Mahalakshmi and Geetha, 2006a). The formula for the cumulative sum of concepts is given by

$$C_{pr} = C \times \left(\sum_{k=0}^{q-1} Q_k \times \left(\sum_{con=0}^{3} Q_{con} \times \left(\sum_{v=0}^{val-1} value_v \right)\right)\right) \tag{11}$$

where, *q* denote the no. of concept qualities present with every concept, Q_{con} denote the quality constraints applicable for every quality of that concept, and *v* denote the no. of values associated with every quality under the given quality constraints. (For example: Concept – crow; quality – color; value: black; quality constraint: mandatory). The quality constraints are three in number. They are: mandatory, optional and exceptional (Mahalakshmi and Geetha, 2006a, Mahalakshmi and Geetha, 2007a).

There can be any number of qualities present with a given concept. But every quality should fall under one of the above quality constraints; if constraint is not mentioned at the time of knowledge inclusion, mandatory is assumed as the default quality constraint. A quality should have one or more values associated with it. The migration of qualities from optional to any of the other three constraints is allowed. But the reverse migration has to be supported with proof. Similarly, the relation is measured by relation priority factor which is given by

$$R_{pr} = R \times \left(\sum_{k=0}^{3} Q_k \times \left(\sum_{con=0}^{4} (Q_{con}) \right)\right) \tag{12}$$

where, *k* denote the no. of relation qualities like invariable concomitance, exclusive, exceptional and direct; *qc* denote the quality constraints of relations like reflexive, symmetric, antisymmetric, asymmetric and transitive; R_{cat} denote the category of relations whether it is one among $R_{S\text{-}OI}$, $R_{S\text{-}R}$, $R_{R\text{-}OI}$ (Mahalakshmi and Geetha, 2006a; Mahalakshmi and Geetha, 2007a).

The KS volunteer who submits the argument evaluates the counter-arguments to identify the knowledge gain. All the other volunteers who respond to the submitted argument, evaluates the argument, which contributes to their respective knowledge gain. The underlying assumption is that, the elements of arguments in the input argument triggers the listening KS volunteers to trace through their own knowledge bases in view of finding the suitable counter-response; thereby, the knowledge units which lie in the trace path are refreshed and re-evaluated, which contributes to their knowledge gain. Therefore, the knowledge gain of the proponent is given by

$$K_{\uparrow pro} = \sum_{k=0}^{w-1} IC(E_A[k]) \, . \tag{13}$$

The respondent prefers to answer the argument only if it finds the details relevant to its' own knowledge base. For finding relevancy, let us assume the entity compares the harvested elements of arguments with that of its own knowledge dictionary. Let the variable *Dictionary* indicate some weight measure, when an element of argument is found in the dictionary. Then, the knowledge gain of the respondent (or opponent) is given by

$$K_{\uparrow res} = \sum_{k=0}^{w-1} (Dictionary \ + IC(E_A[k])) \, . \tag{14}$$

The following section describes the methodology behind RTBSS for knowledge sharing applied in a partially observable non-deterministic argument gaming setting.

5.3 Modified RTBSS Algorithm

The modified RTBSS algorithm for refutation selection is shown in Fig. 5. The algorithm can be well understood by referring to the theory behind in section 3.3 and section 5.1 in this paper. The evaluation should be normalized so that, duplication is avoided. The knowledge base is updated with the output counter-argument. The status of counter-argument with reference to the submitted input argument is hence determined.

Input: max=0;
horizon t;
Ω, the set of all possible observations accumulated as a defect set or hole set; $\Omega = \{h_1, \cdots, h_n\}$
Δ, the set of all possible actions or refutation strategies accumulated as defeat set; $\Delta = \{d_1, \cdots, d_p\}$
Γ, the set of all possible counter-arguments constructed over the recommended action(s) or refutation(s); $\Gamma = \{c_1, \cdots, c_q\}$
Function: Modified RTBSS
//calculate maximum reward
For every element c_q in Γ,
While(t) //Calculate $V_t(b)$ for belief state *b*.
$R(b,d,c,h)=\sum_{s \in S} b(S)R(S)$
$b' = \tau(b,d,c,h)$
$b'(s') = \eta O(s', h, d, c) \sum_{s \in S} T(s,d,c,s')b(s)$
$P(h \mid b, d, c) = \sum_{s' \in S} O(s',h,d,c) \sum_{s \in S} T(s,d,c,s')b(s)$
$V_t(b)=R(b,d,c,h)+\gamma \max_{d \in \Delta} \sum_{h \in \Omega} P(h \mid b, d, c)V_t - 1(\tau(b, d, c, h))$
If(max<V_t(b)) then
max=V_t(b); max_b=b; max_d=d; max_c=c; max_h=h;
end if
end while
end for // apply that counter-argument while gets max reward
$\pi(b)=\arg\max_{d \in \Delta} [R(b,d,c,h)+\gamma \max_{d \in \Delta} \sum_{h \in \Omega} P(h \mid b, d, c)V_t - 1(\tau(b, d, c, h))]$
end function

Figure 5: Modified RTBSS for argument gaming

6 A Running Example

As an explanation, let us look at a sample illustration to demonstrate the art of defect exploration from procedural arguments. The snapshot of knowledgebase of two knowledge sharing entities, KS entity 1 and KS entity 2 before argument exchange are given below (refer Fig. 6 & 7). Let us assume, a KS entity 1 proposes the argument "Mercury is shapeless because it is like water". KS entity 2 inputs the argument, extracts the elements of arguments and labels as subject, reason and object of inference. Therefore, C_S = Mercury; C_R = water; C_{OI} = ! shape.

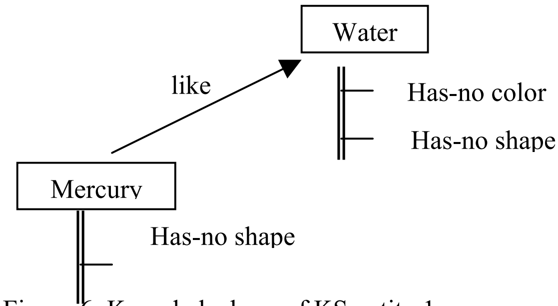

Figure 6: Knowledgebase of KS entity 1

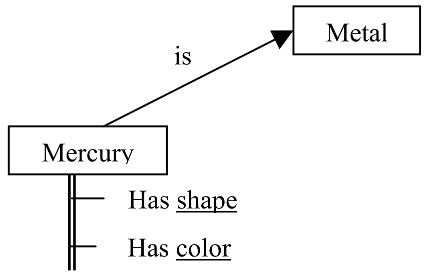

Figure 7: Knowledgebase of KS entity 2

KS Entity 1

A: Mercury is shapeless because it is like water

KS Entity 2

C_S = Mercury; C_R = water; C_{OI} = ! shape

In the above argument scenario, the intention of argument of KS entity 1 was nothing but to prove the shapelessness of 'Mercury', the subject, by stating its' analogousness to 'water', the reason. KS entity 2 now finds that the concept 'Water' does not exists in its very own knowledge base. Therefore, a hole is generated with respect to C_R , ('Water') named, *Concept-non existence.* As per our defect exploration algorithm, knowledge sharing entities cannot derive valid conclusions with incomplete knowledge. In other words, any concept that does not exists in memory, and which is a hindrance to inference, has to be learnt before performing any reasoning via that particular concept. This is shown by step 4 in our algorithm. Hence, to learn about the non-existing concept, KS entity 2 generates an argument "What is water?" as a query to KS entity 1. The response from KS entity 1 is stored in the knowledge base. The resulting knowledgebase of entity 2 is shown in Figure 8.

KS Entity 1

A: Mercury is shapeless because it is like water

A: Water is a substance which has no color.

KS Entity 2

C_S = Mercury; C_R = water; C_{OI} = ! shape

CA: What is water?

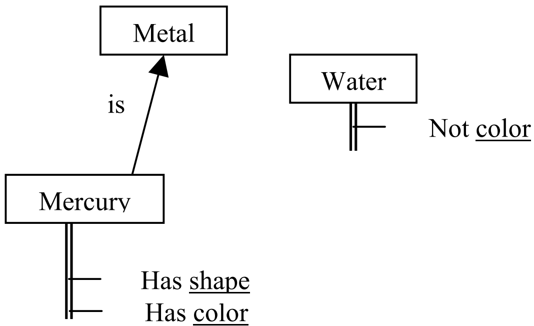

Figure 8: Knowledgebase of KS entity 2 after learning about *concept: water*

After learning the new concept, the hole *Concept-non existence* is now removed and the defect analysis of KS entity 2 again shifts to the argument 1 which discussed about Mercury and its shape. Now, it finds the quality 'shape' which is the negation of C_{OI} of the previous argument, along with a supporting reason, analogousness to 'metal'. Thus, both C_{OI} and !C_{OI} exists, with two valid and equally opposite reasons, and a new hole is generated with respect to C_{OI} as *Antithetical;Concept-disjoint*. Thus, the conflict is due to the presence of two equally valid reasons supporting the existence and non-existence of $C_{OI,}$ the shape of mercury. Therefore, to attack the reason, KS entity 2 now constructs a refutation strategy that possibly generates two different counter-arguments as shown below:

KS Entity 1

A: Mercury is shapeless because it is like water

A: Water is a substance which has no color.

KS Entity 2

C_S = Mercury; C_R = water; C_{OI} = ! shape

CA: What is water?

Refutation(s):

Mercury is not like water because water has no color.

Mercury is not like water because mercury has color

In both the proposed refutations, the relation between C_S and C_R is attacked as "Mercury is not like water" but, the first refutation justifies the quality of C_R and the second refutation justifies the quality of C_S. But the best part would be to oppose C_R. Therefore, that refutation which explains more about the nature of C_R is taken as the best refutation strategy.

KS Entity 1 **KS Entity 2**

A: Mercury is shapeless because it is like water C_S = Mercury; C_R = water; C_{OI} = ! shape

CA: What is water?

A: Water is a substance which has no color.

CA: Mercury is not like water because water has no color.

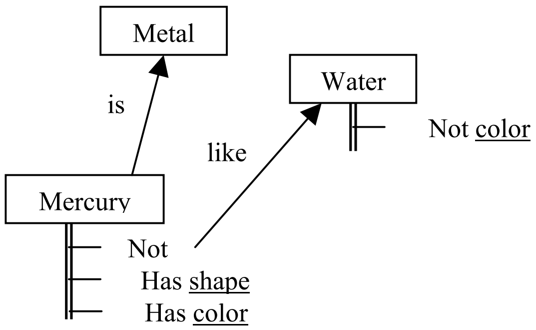

Figure 9: Enhanced knowledgebase of KS entity 2

After the decision about best refutation, the FSA for counter-arguments constructs the output counter-argument as "Mercury is not like water because water has no color" and updates its own knowledge base. The self-learning of KS entity is shown by the relation "not like" between concepts "Mercury" and "Water" in Fig. 9. But, however, KS entity 1 finds analogousness between "Mercury" and "Water" because of "no shape". Therefore the next argument from KS entity 1 would be something around the *quality: shape* of "Mercury". But this time, it would carefully withdraw the reason why "Mercury" is shapeless. All it knows about the shapelessness of "Mercury" is that, "Mercury" is analogous to "Water" and so, it possess no shape. But this reason has been attacked previously by the KS entity 2. Therefore, the argument without any reason is generated.

KS Entity 1 **KS Entity 2**

A: Mercury is shapeless because it is like water C_S = Mercury; C_R = water; C_{OI} = ! shape

CA: What is water?

A: Water is a substance which has no color.

CA: Mercury is not like water because water has no color.

A: But, Mercury is shapeless.

The word, "But" in the argument states that similar argument has already been proposed for which satisfiable answers were not received. Of course, this is true, because, our KS entity 2 attacked only the reason and not the object of inference, the *shape of Mercury*. After finding the repeated argument of entity 1, entity 2 now infers a conflict of properties between the concept :Mercury of the argument and concept :Mercury of its own knowledgebase. The hole *Antithetical;Concept disjoint* is generated with respect to the *quality:shape* of *concept:Mercury.* But here, the analysis on supporting reason of entity 1's argument is useless because, it has been disproved previously. Therefore, it has to be recorded that KS entity 1 has no further information to present as supporting reason with its argument and this is an indication for KS entity 2 to take its own course of discussion. Now, the KS entity 2 generates a new information concentrating around the shape of Mercury and generates a new counter-argument at the output.

KS Entity 1 **KS Entity 2**

A: Mercury is shapeless because it is like water C_S = Mercury; C_R = water; C_{OI} = ! shape

CA: What is water?

A: Water is a substance which has no color.

CA: Mercury is not like water because water has no color.

A: But, Mercury is shapeless.

CA: Mercury has shape because it is a metal.

KS entity 1 does not know anything about Metals. Therefore, temporarily the discussion comes to a halt until KS entity 1 learns about Metals and liquid metals and the classification of Mercury as a liquid metal. After that, the discussion would again re-gain its own course, thus teaching the entity 2 like: Mercury is a metal. Mercury falls into a new category called liquid metal. Let us assume, the KS entity 2 does not have any knowledge about shape (refer Figure 10). Then, with our first argument proposed by KS entity 1, two holes are generated within the defect analysis of KS entity 2, one for opposing C_{OI}:'shape' and other for opposing CR: 'water'. Then it is up to the entity to identify the best hole, here, C_{OI}:'shape' which might fall into the defect category "unestablished to subject". Therefore, appropriate refutation is constructed to cover the generated hole, and the counter-argument "Mercury has shape because it is a metal" is generated.

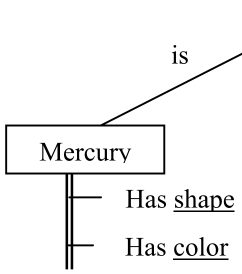

Figure 10: Alternate Knowledgebase of KS entity 2

KS Entity 1	KS Entity 2
A: Mercury is shapeless because it is like water	C_S = Mercury; C_R = water; C_{OI} = ! shape
	CA: Mercury has shape because it is a metal.

Further assumptions of concepts of knowledgebase would give rise to more holes and eventually more defeat strategies, out of which the best one is given out as output counter-argument. The above example is not suitable for illustrating a proof with respect to the relation of type: invariable, existing between CR and COI. However, further arguments stating "Mercury is a metal which is exceptionally in liquid state" may introduce exceptional type of relations for the CS, Mercury. Thus, every output counter-argument will carry the discussion in different directions, and hence, it is for the discussing entities to stop discussion when they reach a common agreed upon conclusion. Thus new beliefs about the shape of mercury and its analogousness to every other concept are being learnt by both the entities during knowledge sharing.

Experimental SetUp:

For implementation, we have assumed the Birds domain with three entities participating in the discussion. The rewards obtained by each of those agents are projected in Figure 11. For discussion, the agents exchange various arguments related to 'Birds' domain. Reward assignment is of two types: (a) rewarding the opponent (b) rewarding the self. Rewarding the opponent means assigning rewards to the opponent for their behaviour during discussion. Behavior, here, we mean, the accuracy and utility factor of the knowledge shared from that opponent. Rewarding the self is like encouraging oneself for having performed better in sharing one's own knowledge to the opponent. In other words, this is a measure on how a particular agent is able to co-operate and understand the arguments from the opponent. It is like 'patting one's own back!!' The objective behind this kind of rewards is to project one's ability in a group.

In Fig.11, every agent (x-axis) participates in the discussion. Therefore, every agent is expected to have 3 reward values (because there are 3 members discussing with each other); one reward for self and two other reward entries for the colleagues in the community. Fig.11 (a) depicts the reward scenario of all the three agents before the start of discussion about 'Birds' domain. (Note: This is an intermediate scenario. The values are taken as a result of previous discussions).

Agent C has not participated in the discussion and the rewards for agent *C* has no change at the output. It can be seen that the rewards for discussing entities Agent *A* and Agent *B* improves considerably indicating the amount of knowledge learnt by both of the entities. i.e. the reward value for Agent *A* in Agent *B* (x-axis) in Fig. 11 (b) is increased than that in Fig. 11 (a). This is shown by a blue circle highlighting the changes in both the graphs. Also, the reward value for Agent *B* in Agent *A* (x-axis) in Fig. 11 (b) is increased when compared to the same situation in Fig. 11 (a). This is shown by a red circle highlighting the changes in both the graphs. The communication between the volunteers is achieved by using Indian logic based ontologies as part of their knowledge representation. The key idea is to have volunteers that teach a new or unknown information or concept to interested volunteers. Every participating knowledge volunteers performs reasoning and inference over the knowledge obtained which is reflected at the output through cumulative rewards.

7 Limitations in PONAG

The proposed architecture of PONAG system for knowledge sharing assumes two participating entities, which is modeled after the teaching-learning system of knowledge sharing. Practically, there can be more number of learners to one teacher, every one of them communicating asynchronously in independent sessions with the teacher (Rybinski and Ryzko, 2003). The RTBSS (Real Time Belief Space Search) functions for deciding optimal policy, π and Reward, R can be made useful in making online decisions about the selection of output counter-arguments. Interactive POMDPs (Gymtrasewicz and Doshi, 2003) are better suitable here but modelling I-POMDPs into the proposed PONAG is not sufficient and will surely result in inference conflicts due to the lack of versioning of knowledge base for every independent KS session. Moreover, not every inference made is reflected at the immediate refutation. Certain inferences may be reflected at the output only after a considerable no. of argument exchanges. Though the procedural nature of argumentation takes care of the above issue by evaluating the inferences made at every stage of argument exchange, utilization of such previously made inferences in constructing refutations at a future stage of discussion should be taken into account while evaluating the counter-argument of a later stage of discussion. In future, we plan to devise a knowledge sharing strategy where the sharing of knowledge will be based on confidentiality of knowledge units of the volunteers and mutual trust (Harish *et al.*, 2007).

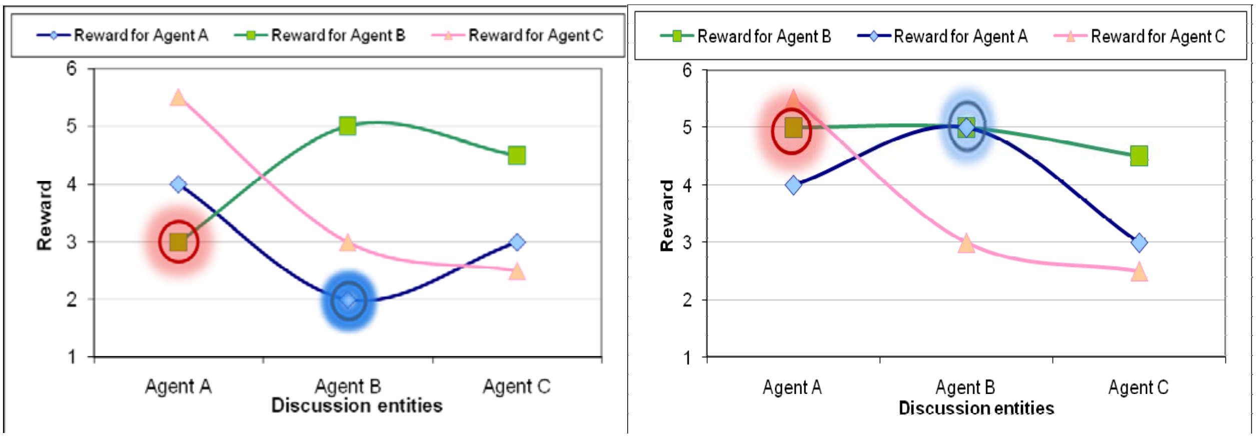

Figure 11: (a) Reward before discussion (b) Reward after discussion

8 Conclusions

The paper proposed the design and implementation of construction of better counter-arguments through the policy of refutation selection. Assigning of rewards were the major contributing factor for deciding upon the efficient counter-argument. The efficient counter-response depended on the nature and amount of holes that a response could cover. The reflection of knowledge exchange is clearly visible in the amount of inference and the revision of knowledge units. The nature of partial observability of argument gaming setting is the main criterion for analyzing the arguments in detail. Because the knowledge exchange has to be rational, allowing certain amount of partial observability tends to preserve the rational exchange of arguments between the participating entities. In another perspective, it can be concluded that, due to partial observability the belief search algorithms should be more strengthened to obtain better counter-arguments. Aiming higher, we have not considered the computational perspectives (time and complexity) of the algorithms. In future, an optimal decision making algorithm for refutation selection has to be designed including the complexity constraints.

References

[1] Aghila, G., G.S. Mahalakshmi, and T.V. Geetha, KRIL – a knowledge representation system based on Nyaya Shastra using extended description logics, *VIVEK Journal*, vol.15, no.3, pp.3-18, 2003.

[2] Atkinson, K., T.B. Capon, and P.M. Burney, A dialogue game protocol for multi-agent argument over proposals for action, *Journal of Autonomous Agents and Multi-Agent Systems*, Springer Science + Business Media, Inc., 2005.

[3] Burney, P.M., and S. Parsons, *Dialogue Game Protocols*, In Marc-Philippe Huget, editor, Agent Communications Languages, Berlin, Germany, Springer, 2003.

[4] Bench-Capon, T.J.M., Representation of case law as an argumentation framework in T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), *Legal Knowledge and Information Systems, Jurix 2002, The Fifteenth Annual Conference*, Amsterdam, IOS Press, pp.103-112, 2002.

[5] Carbogim, D.V., D. Robertson, and L. John, *Argument Based Applications to Knowledge Engineering*, Division of Informatics, University of Edinburgh, Brazil, 1999.

[6] Chakraborti, C., *Logic: Informal, Symbolic and Inductive*, Prentice Hall of India, New Delhi, 2006.

[7] Irene, Y., L. Chen, and S.J.H. Yang, Peer-to-Peer knowledge sharing (KS) in the collaboration supported virtual learning communities, *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, 2006.

[8] Gautama (Gotama), *The Nyaya Sutras*, translated by S.C. Vidyabhusana, edited by Nanda Lal Sinha, Sacred Book of the Hindus, Allahabad, 1930, Reprinted in 1990, Delhi: Motilal Banarsidass

[9] Geffner, H., and B. Bonet, *Solving Large Pomdps Using Real Time Dynamic Programming*, 1998.

[10] Gradinarov, P.I., *Phenomenology and Indian Epistemology - Studies in Nyaya-Vaisesika Transcendental Logic and Atomism*, Sophia Indological Series, 1990.

[11] Gymtrasiewicz, P., and P. Doshi, Interactive POMDPs: properties and preliminary results, *Proceedings of International Conference on AAMAS'04*, 2004.

[12] Harish, M., G.S. Mahalakshmi, and T.V. Geetha, Design and analysis of a game theoretic model for P2P trust management, *Proc. of 4th International Conf. on Distributed Computing and Internet Technology*, LNCS (Springer-Verlag), Bangalore, India, pp.17-20, 2007.

[13] Hoey, J., A.V. Bertoldi, *et al.*, Assisting persons with dementia during handwashing using a partially observable markov decision process, *Proceedings of the International Conference on Vision Systems*, 2007.

[14] Ilavarasu, S., *Nannool Ezhuthathikaaram*, Manivasagar Publishers, Chennai, 1999.

[15] King, R., *Indian Philosophy – An Introduction to Hindu and Buddhist Thought*, EUP, Indian Edition: Maya Publishers Pvt. Ltd., New Delhi, India, 1999.

[16] Mahalakshmi, G.S., G. Aghila, and T.V. Geetha, Multi-level ontology representation based on Indian logic system, *4th International Conference on South Asian Languages ICOSAL-4*, Chidambaram, India, October 2002.

[17] Mahalakshmi, G.S., and T.V. Geetha (2006a), A mathematical model for argument procedures based on Indian philosophy, *Proc. of International Conf. Artificial Intelligence and Applications as part of the 24th IASTED International Multi-conference Applied Informatics*, Innsbruck, Austria, 2006.

[18] Mahalakshmi, G.S., and T.V. Geetha (2006b), Architecture of Indian-logic based procedural argumentation system for knowledge sharing, *Proc. of IEEE Systems, Man & Cybernetics Society 5th Chapter International Conf. on Advances in Cybernetic Systems*, Sheffield Hallam University, UK, 2006.

[19] Mahalakshmi, G.S., and T.V. Geetha (2007a), Navya-Nyaya Approach to Defect Exploration in Argument Gaming for Knowledge Sharing, *Proceedings of International Conference On Logic, Navya-Nyaya & Applications - A Homage To Bimal Krishna Matilal*, Jadavpur University, Calcutta, India, 2007.

[20] Mahalakshmi, G.S., and T.V. Geetha (2007b), I-KARe - a rational approach to knowledge acquisition and reasoning using Indian logic based knowledge models, *Proceedings of the 3rd Indian International Conference on Artificial Intelligence*, Pune, India, 2007.

[21] Mahalakshmi, G.S., and T.V. Geetha (2007c), The logic of reasoning by procedural argumentation for knowledge sharing, *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, India, IEEE CS Press, 2007.

[22] Mahalakshmi, G.S., and T.V. Geetha (2008a), An algorithm for knowledge sharing by procedural argument representations based on Indian philosophy, *International Journal of Computer, Mathematical Sciences and Applications*, vol.1, no.3-4, pp.253-267, 2008.

[23] Mahalakshmi, G.S., and T.V. Geetha (2008b), Gurukulam-reasoning based learning system using extended description logics, Dr. Estrella Pulido and Dr. Maria D. R-Moreno eds., *International Journal of Computer Science and Applications - Special Issue on New Trends on AI techniques for Educational Technologies*, Technomathematics Research Foundation , vol.5, no.2, pp.14 – 32, 2008.

[24] Mahalakshmi, G.S., and T.V. Geetha (2008c), An Indian logic-based knowledge-sharing architecture for virtual knowledge communities, *International Journal of Networking and Virtual Organisation–special edition: Virtual Learning and Knowledge Sharing*, Inderscience Publishers, 2008.

[25] Nordquist, R., *Grammar & Composition*, http://grammar.about.com/, 2008.

[26] Oetke, C., Indian logic and Indian syllogism, *Indo-Iranian Journal*, vol.46, no.1, pp.53-69(17), Kluwer Academic Publishers, 2003.

[27] Paquet, S., L. Tobin, and B. Chaib-draa (2005a), An online POMDP algorithm for complex multiagent environments, *Proceedings of International Conference on AAMAS'05*, Netherlands, 2005.

[28] Paquet, S., L. Tobin, and B. Chaib-draa (2005b), Real-time decision making for large POMDPs, *Canadian Conference on AI*, pp.450-455, 2005.

[29] Parsons, S., and P.M. Burney, Argumentation based communication between agents, M.P.Huget (Ed.): *Communication in Multi-agent Systems*, LNAI 2650, pp.164-178, 2003.

[30] Porter, D., *Argumentation and Debate*, New York, 1954.

[31] Prakken, H., and S. Renooij, Reconstructing causal reasoning about evidence: a case study, in Bart Verheij, Arno R. Lodder, Ronald P. Loui and Antoinette J. Muntjwerff (eds.), *Legal Knowledge and Information Systems Jurix 2001: 14th Annual Conference*, Amsterdam: IOS Press, pp.131-137, 2001.

[32] Prakken, H., Coherence and flexibility in dialogue games for argumentation, *Oxford University Press*, vol.15, no.6, 2005.

[33] Roos, N., On resolving conflicts between arguments, *Computational Intelligence*, vol.16, no.3, 2000.

[34] Rybinski, H., and D. Ryzko, Knowledge sharing in default reasoning based multi-agent systems, *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, 2003.

[35] Ballantyne, J.R., Lectures on the Nyaya Philosophy- Embracing the Text of Tarka Samgraha, Presbyterian Mission Press, Allahabad, India, 1849, digitized by Google Books on the web at http://books.google.com

[36] Iyer, S.U.Ve., *Pavanandhi Munivar Iyatria Nannool Moolamum Mayilainaathar Uraiyum*, Dr.U.Ve.Saaminaatha Iyer Noolnilaiyam, Besant Nagar, Chennai, 1995.

[37] Solomon, E., *Indian Dialectics: Methods of Philosophical Discussion*, 2 vols, Ahmedabad: Inst. of Learning and Research, Gujarat Vidya Sabha, India, 1976.

[38] Spohn W., *A Brief Comparison of Pollock's Defeasible Reasoning and Ranking Functions*, Konstanz University, Germany, 2002.

[39] vaaraNanaar, K.V., *Tolkaapiyam – Nannool Ezhuthathikaaram*, Annamalai University, Tamilnadu, India, 3rd Edition, 1974.

[40] Verheij, B., Artificial argument assistants for defeasible argumentation, Dept. of Metajurisdica, University, Masstricht, Netherlands, 2003.

[41] Vidyabhusana, S.C., *A history of Indian logic – Ancient, Medieaeval and Modern schools*, Motilal Banarsidass Publishers Private Ltd., Delhi, India, 1988.

[42] Virupakshananda, S., Tarka Samgraha and Sri Ramakrishna Math, Madras, 1994.

[43] Vreeswijk, G., and J. Hulstijn, A free-format dialogue protocol for multi-party inquiry, *Proc. of the 8th International Workshop on the Semantics and Pragmatics of Dialogue*, pp.273-279, 2004.

[44] Walton, D.N., and E.C.W. Krabbe, Commitment in dialogue: basic concepts of interpersonal reasoning, *SUNY Press*, Albany, NY, pp.165-167, 1995.

[45] Walton, D., Justification of argumentation schemes, *Australasian Journal of Logic (3)*, 2005.

[46] Yamashita, M., Y. Umeda, and H. Sawamura, Applications of the argument-based agent system with dialectical reasoning capability, *Proceedings of Fourth International Conference on Knowledge-based Intelligent Engineering Systems & Allied Technologies*, UK, 2000.