Argumentation strategies for plan resourcing

Chukwuemeka D. Emele
Dept. of Computing Science
University of Aberdeen,
Aberdeen, AB24 3UE, UK
c.emele@abdn.ac.uk

Timothy J. Norman
Dept. of Computing Science
University of Aberdeen,
Aberdeen, AB24 3UE, UK
t.j.norman@abdn.ac.uk

Simon Parsons
Comp. & Infor. Science Dept.
Brooklyn College, CUNY,
Brooklyn, 11210 NY, USA
parsons@sci.brooklyn.cuny.edu

ABSTRACT

What do I need to say to convince you to do something? This is an important question for an autonomous agent deciding whom to approach for a resource or for an action to be done. Were similar requests granted from similar agents in similar circumstances? What arguments were most persuasive? What are the costs involved in putting certain arguments forward? In this paper we present an agent decision-making mechanism where models of other agents are refined through evidence from past dialogues, and where these models are used to guide future argumentation strategy. We empirically evaluate our approach to demonstrate that decision-theoretic and machine learning techniques can both significantly improve the cumulative utility of dialogical outcomes, and help to reduce communication overhead.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Experimentation

Keywords

Argumentation, Decision theory, Machine learning, Policies

1. INTRODUCTION

It is typically the case that collaborative activities require agents (human or artificial) to share resources, act on each others' behalf, coordinate individual acts, etc. Agreements to collaborate are often *ad-hoc* and temporary in nature but can develop into more permanent alliances. Regardless of whether such relationships are transient or permanent, dialogue among collaborators that is concerned with the delegation of tasks, or sharing of resources are common.

The formation of agreements may, however, be subject to policy (or norm) restrictions. Such policies might regulate what resources may be released to a partner from some other organisation, under what conditions they may be used, and what information regarding their use is necessary to make a decision. Similarly, policies may govern actions that can be done either to pursue personal goals or on behalf of another.

One important aspect of collaborative activities is resource sharing and task delegation [3]. If a plan is not properly resourced and tasks delegated to appropriately competent agents then collaboration may fail to achieve shared goals. We explore in this paper strategies for plan resourcing where agents operate under policy constraints. This is important not only for autonomous agents operating on behalf of individuals or organisations, but also if these agents support human decision makers in team contexts. To guide strategies regarding whom to approach for a resource and what arguments to put forward to secure an agreement, agents require accurate models of other decision makers that may be able to provide such a resource. The first question addressed in this research is how we may utilise evidence from past encounters to develop accurate models of the policies of others (Section 2).

Given that agents are operating under policies, and some policies may prohibit an agent from providing a resource to another under certain circumstances, how can we utilise the model of others' policies that have been learned to devise a strategy for selecting an appropriate provider from a pool of potential providers? To do this, we propose a decision-theoretic model, which utilises a model of the policies and resource availabilities of others to aid in deciding who to talk to and what information needs to be revealed if some other collaborator is to provide a resource (Section 3).

In this paper, we demonstrate the utility of our approach by testing the following hypotheses: (i) decision-theoretic and machine learning techniques can significantly improve the cumulative utility of dialogical outcomes; and (ii) this combination of techniques can help to focus dialogue on pertinent issues for negotiation (Section 4).

2. LEARNING POLICIES

The framework we propose here (illustrated in Figure 1) enables agents to negotiate regarding resource provision, and use evidence derived from argumentation to build more accurate and stable models of others' policies. These policy models, along with models of resource availability also derived from previous encounters, are used to guide dialogical strategies for resourcing plans. The dialogue manager handles all communication with other agents. In learning policies from previous encounters, various machine learning techniques can be employed; Figure 1 refers to a rule learning mechanism, but we also investigate instance-based and decision-tree learning in this paper (Section 2.3). The ar-

Cite as: Argumentation strategies for plan resourcing, Chukwuemeka D. Emele, Timothy J. Norman and Simon Parsons, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 913-920.

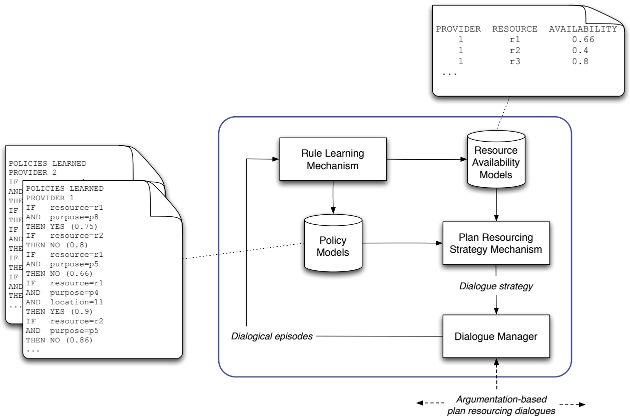

Figure 1: Agent reasoning architecture

guments exchanged during dialogue constitute the evidence used to learn policies and resource availability. Arguments refer to features of the task context in which a resource is to be used, and decisions regarding whether or not a resource is made available to another agent may depend on such features. The plan resourcing strategy mechanism reasons over policy and resource availability models, and uses decision theoretic heuristics to select which potential provider yields the highest expected utility (see Section 3). In order to model our argumentation-based framework, we begin by formulating a mechanism to capture policies.

2.1 Policies

Agents have policies (aka. norms) that govern how resources are provided to others. In our model, policies are conditional; they are relevant to an agent's decision under specific circumstances. These circumstances are characterised by a set of features. Some examples of features may include: (1) the height of a tower, (2) the temperature of a room, or (3) the manufacturer of a car.

Definition 1 ***(Features)*** Let $\mathcal{F}$ be the set of all features such that $f_1, f_2, \ldots \in \mathcal{F}$. We define a feature as a characteristic of the prevailing circumstance under which an agent is operating (or carrying out an activity); i.e. the task context.

Our concept of policy maps a set of features into an appropriate policy decision. In our framework, an agent can make one of two policy decisions, namely (1) *grant*, which means that the policy allows the agent to provide the resource when requested, and (2) *deny*, which means that the policy prohibits the agent from providing the resource.

Definition 2 ***(Policies)*** A policy is defined as a function $\Pi : \vec{\mathcal{F}} \rightarrow \{grant, deny\}$, which maps feature vectors of tasks, $\vec{\mathcal{F}}$, to appropriate policy decisions.

In order to illustrate the way policies are captured in this model, we present the following examples (see Table 1). Assuming, f_1 is resource, f_2 is purpose, f_3 is weather report (with respect to a location), f_4 is the affiliation of the agent, and f_5 is the day the resource is required then policies $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ (see Table 1) will be interpreted as follows:

$\mathbb{P}_1$**:** You are **permitted** to release a *helicopter* (h), to an agent if the *helicopter* is required for the purpose of transporting relief materials (trm).

Table 1: An example policy profile.

Policy Id	f_1	f_2	f_3	f_4	f_5	Decision
$\mathbb{P}_1$	h	trm				grant
$\mathbb{P}_2$	h		vc			deny
$\mathbb{P}_3$	j					grant
$\mathbb{P}_4$	c		vc	xx		grant
...	...	...	...	...	...	...
$\mathbb{P}_n$	q	yy	w	xx	z	deny

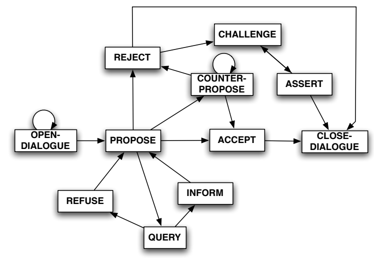

Figure 2: The negotiation protocol.

$\mathbb{P}_2$**:** You are **prohibited** from releasing a *helicopter* to an agent if the weather report says there are volcanic clouds (vc) in the location the agent intends to deploy the *helicopter*.

$\mathbb{P}_3$**:** You are **permitted** to release a jeep (j) to an agent.

If a *helicopter* is intended to be deployed in an area with volcanic clouds then the provider is forbidden from providing the resource but might offer a ground vehicle (e.g. *jeep*) to the seeker if there is no policy prohibiting this and the resource is available.

2.2 Argumentation-based Negotiation

The protocol employed in this framework, constraining dialogical moves, is illustrated in Figure 2. Our approach in this regard is similar to the dialogue for resource negotiation proposed by McBurney & Parsons [4].

To illustrate the sorts of interaction between agents, consider the example dialogue in Figure 3. Let x and y be seeker and provider agents respectively. Suppose we have an argumentation framework that allows agents to ask for and receive explanations, offer alternatives, or ask for more information about the attributes of requests, then there is the potential for x to gather additional evidence regarding the likely policy rules guiding y concerning provision of resources.

Negotiation for resources takes place in a turn-taking fashion. The dialogue starts when x sends a request (propose in Figure 2) to y (e.g. line 1, Figure 3). The provider, y, may respond by conceding to the request (accept), rejecting it, offer an alternative resource (counter-propose), or ask for more information (query) such as in line 2 in Figure 3. If the provider agrees to provide the resource then the negotiation ends. If, however, the provider rejects the proposal (line 8, Figure 3), then the seeker may challenge that decision (line 9), and so on. If the provider suggests an alternative then the seeker evaluates it to see whether it is acceptable or not. Furthermore, if the provider agent needs more information from the seeker in order to make a decision, the

#	Scenario
1	x: Can I have a *helicopter* for \$0.1M reward?
2	y: What do you need it for?
3	x: To transport relief materials.
4	y: To where?
5	x: A refugee camp near Indonesia.
6	y: Which date?
7	x: On Friday 16/4/2010.
8	y: No, I can't provide you with a *helicopter*.
9	x: Why?
10	y: I am not permitted to release a *helicopter* in volcanic eruption.
11	x: There is no volcanic eruption near Indonesia.
12	y: I agree, but the ash cloud is spreading, and weather report advises that it is not safe to fly on that day.
13	x: Ok, thanks.
14	y: You're welcome.

Figure 3: Dialogue example.

provider agent would ask questions that will reveal the features it requires to make a decision (query, inform/refuse). There is a cost attached to the revelation of private information to other agents. An agent might refuse to reveal a piece of information if doing so is expensive [7], and this may vary depending upon who it is revealed to. The negotiation ends when agreement is reached or all possibilities explored have been rejected.

Furthermore, since we make the simplifying assumption that agents communicate truthfully and accurately in this framework[1], the suggestion of an alternative by a provider could serve as evidence that the provider agent does not have any policy that forbids the provision of such a resource to the seeker, and that the resource is also available.

2.3 Learning from dialogue

One of the core goals of this research is to learn models of the policies of others. When an agent has a collection of experiences with other agents described by feature vectors (see Section 2.1), we can make use of existing machine learning techniques for learning associations between sets of discrete attributes (i.e. elements of $\mathcal{F}$) and policy decisions. Specifically, we investigate three types of machine learning algorithms[2] [5], namely decision tree learning (using C4.5), instance-based learning (using k-nearest neighbour), and rule-based learning (using sequential covering). Figure 4 shows an example decision tree representing a model of the policies of some other agent learned from interactions with that agent. Nodes of the decision tree capture features of an agent's policy, edges denote feature values, while the leaves are policy decisions. Similarly, the policy models illustrated in Figure 1 show the kind of rules learnt using sequential covering.

The three machine learning algorithms investigated here have very different properties. Instance-based learning is useful in this context because it can adapt to and exploit evidence from dialogical episodes as they accrue. In contrast, decision trees and rule learning are not incremental; the tree or the set of rules must be reassessed periodically as new evidence is acquired. We define a *learning interval*,

[1]While the issue of deception remains an open problem, some techniques for addressing this assumption have been investigated [8].

[2]We use the Weka [10] implementation of these algorithms.

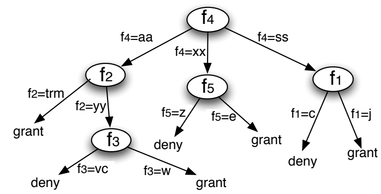

Figure 4: Example decision tree.

ϕ, which determines the number of interactions an agent must engage in before building (or re-building) its policy model. Once an agent has had ϕ interactions, the policy learning process proceeds as follows. For each interaction, which involves resourcing a task t using provider y, we add the example $(\vec{F}_y, grant)$ or $(\vec{F}_y, deny)$ to the training set, depending on the evidence obtained from the interaction. The model is then constructed. In this way, an agent may build a model of the relationship between observable features of agents and the policies they are operating under. Subsequently, when faced with resourcing a new task, the policy model can be used to obtain a prediction of whether a particular provider has a policy that permits the provision of the resource.

Learning mechanisms such as sequential covering have a number of advantages over instance-based approaches; in particular, the rules (or trees) learnt are more amenable to scrutiny by a human decision maker.[3] It should be noted, however, that the framework presented here is agnostic to the machine learning mechanism employed.

We also adopt a simple, off the shelf, probabilistic approach to compute the probability of a resource being available based on past experience, but there are far more sophisticated approaches to model resource availability; e.g. [2].

3. ARGUMENTATION STRATEGIES

Having described how the policies of others can be learned with the help of evidence derived from argumentation, here we demonstrate the use of such structures in developing argumentation strategies for deciding which agent to negotiate with and what arguments to put forward. Our model takes into account communication cost and the benefit to be derived from fulfilling a task. Agents attempt to complete tasks by approaching the most promising provider. Here, we formalise the decision model developed for this aim; a model that we empirically evaluate in Section 4.

Let $\mathcal{A}$ be a society of agents. In any encounter, agents play one of two roles: seeker or provider. Let $\mathcal{R}$ be the set of resources such that $r_1, r_2, \ldots \in \mathcal{R}$ and $\mathcal{T}$ be the set of tasks such that $t_1, t_2, \ldots \in \mathcal{T}$, and, as noted above, $\mathcal{F}$ is the set of features of possible task contexts. Each seeker agent $x \in \mathcal{A}$ maintains a list of tasks $t_1, t_2, \ldots t_n \in \mathcal{T}$ and the rewards $\Omega_x^{t_1}, \Omega_x^{t_2}, \ldots \Omega_x^{t_n}$ to be received for fulfilling each corresponding task. We assume here that tasks are independent; in other words, x will receive $\Omega_x^{t_1}$ if t_1 is fulfilled irrespective of the fulfilment of any other task. Further, we assume that tasks require single resources that can each be provided by a single agent; i.e. we do not address problems

[3]Sequential covering does not necessarily find the best or smallest set of rules, but other, more sophisticated rule induction methods may equally be employed [5].

related to the logical or temporal relationships among tasks or resources. Providers operate according to a set of policies that regulate its actions, and (normally) agents act according to their policies. For example, a car rental company may be prohibited from renting out a car if the customer intends to travel across a country border.

Each seeker agent $x \in \mathcal{A}$ has a function μ_x^r with signature $\mathcal{A} \times \mathcal{R} \times \mathcal{T} \times 2^{\mathcal{F}} \rightarrow \mathbb{R}$ that computes the utility gained if x acquires resource $r \in \mathcal{R}$ from provider $y \in \mathcal{A}$ in order to fulfil task $t \in \mathcal{T}$, assuming that the information revealed to y regarding the use of r is $F \subseteq \mathcal{F}$. This F will typically consist of the information features revealed to persuade y to provide r within a specific task context. (Although we focus here on resource provision, the model is equally applicable to task delegation, where we may define a function $\mu_x^t : \mathcal{A} \times \mathcal{T} \times 2^{\mathcal{F}} \rightarrow \mathbb{R}$ that computes the utility gained if y agrees to complete task t for x, assuming that the information revealed to y to persuade it to do t is $F \subseteq \mathcal{F}$.)

Generally, agents receive some utility for resourcing a task and incur costs in providing information, as well as paying for the resource. In some domains, there may be other benefits to the seeker and/or provider in terms of some kind of non-monetary transfers between them, but we do not attempt to capture such issues here. Hence, in our case, the utility of the seeker is simply the reward obtained for resourcing a task minus the cost of the resource and the cost of revealing information regarding the task context.

Definition 3 ***(Resource Acquisition Utility)*** The utility gained by x in acquiring resource r from y through the revelation of information F is:

$$\mu_x(y, r, t, F) = \Omega_x^t - (\Phi_y^r + Cost_x(F, y))$$

where Ω_x^t is the reward received by x for resourcing task t, Φ_y^r is the cost of acquiring r from y (which we assume to be published by y and independent of the user of the resource), and $Cost_x(F, y)$ is the cost of revealing the information features contained in F to y (which we define below).

The cost of revealing information to some agent captures the idea that there is some risk in informing others of, for example, details of private plans.

Definition 4 ***(Information Cost)*** We model the cost of agent x revealing a single item of information, $f \in \mathcal{F}$, to a specific agent, $y \in \mathcal{A}$, through a function: $cost_x : \mathcal{F} \times \mathcal{A} \rightarrow \mathbb{R}$. On the basis of this function, we define the cost of revealing a set of information $F \in \mathcal{F}$ to agent y, as the sum of the cost of each $f \in F$.

$$Cost_x(F, y) = \sum_{f \in F} cost_x(f, y)$$

Cost, therefore, depends on y, but not on the task/resource. This definition captures a further assumption of the model; i.e. that information costs are additive. In general, we may define a cost function $Cost'_x : 2^{\mathcal{F}} \times \mathcal{A} \rightarrow \mathbb{R}$. Such a cost function, however, will have some impact upon the strategies employed (e.g. if the cost of revealing f_j is significantly higher if f_k has already been revealed), but the fundamental ideas presented in this paper do not depend on this additive information cost assumption.

Predictions regarding the information that an agent, x, will need to reveal to y for a resource r to persuade it to make that resource available is captured in the model that x has developed of the policies of y. For example, if, through prior experience, it is predicted that a car rental company will not rent a car for a trip outside the country, revealing the fact that the destination of the trip is within the country will be necessary. Revealing the actual destination may not be necessary, but the costs incurred in each case may differ. Let $Pr(Permitted|y, r, F)$ be the probability that, according to the policies of y (as learned by x), y is permitted to provide resource r to x given the information revealed is F.

Predictions about the availability of resources also form part of the model of other agents; e.g. the probability that there are cars for rent. Let $Pr(Avail|y, r)$ be the probability of resource r being available from agent y. These probabilities are captured in the models learned about other agents from previous encounters.

Definition 5 ***(Resource Acquisition Probability)*** A prediction of the likelihood of a resource being acquired from an agent y can be computed on the basis of predictions of the policy constraints of y and the availability of r from y:

$$Pr(Yes|y, r, F) = Pr(Permitted|y, r, F) \times Pr(Avail|y, r)$$

With these definitions in place, we may now model the utility that an agent may expect to acquire in approaching some other agent to resource a task.

Definition 6 ***(Expected Utility)*** The utility that an agent, x, can expect by revealing F to agent y to persuade y to provide resource r for a task t is computed as follows:

$$E(x, y, r, t, F) = \mu_x(y, r, t, F) \times Pr(Yes|y, r, F)$$

At this stage we again utilise the model of resource provider agents that have been learned from experience. The models learned also provide the minimal set of information that needs to be revealed to some agent y about the task context in which some resource r is to be used that maximises the likelihood of there being no policy constraint that restricts the provision of the resource in that context. This set of information depends upon the potential provider, y, the resource being requested, r, and the task context, t. (If, according to our model, there is no way to convince y to provide the r in context t, then this is the empty set.)

Definition 7 ***(Information Function)*** The information required for y to make available resource r in task context t according to x's model of the policies of y is a function $\lambda_x : \mathcal{A} \times \mathcal{R} \times \mathcal{T} \rightarrow 2^{\mathcal{F}}$

Now, we can characterise the optimal agent to approach for resource r, given an information function λ_x as the agent that maximises the expected utility of the encounter:

$$y_{opt} = \arg\max_{y \in \mathcal{A}} E(x, y, r, t, F) \text{ s.t. } F = \lambda_x(y, r, t)$$

Our aim here is to support decisions regarding which agent to approach regarding task resourcing (or equivalently task performance); an aim that is met through the identification of y_{opt}. The question remains, however, how the agent seeking a resource presents arguments to the potential provider, and what arguments to put forward. To this aim, we present argumentation strategies that focus on minimising communication overhead (i.e. reducing the number of messages between agents) and minimising the information communicated (i.e. reducing the cost incurred in revealing information). To illustrate these strategies, consider a situation in which, according to the evaluation made by x (the seeker) of y_{opt}'s (the provider's) policies, $\lambda_x(y_{opt}, r, t) = \{f_1, f_2, f_3, f_4\}$ for resource r used for task t. The costs for revealing each feature is, as described above, $cost_x(f_1, y_{opt})$, etc. Using this situation, in the following sections we discuss 3 strategies: message minimisation; profit maximisation; and combined.

3.1 Message minimisation

The rationale for the use of this first strategy is for the seeker agent, x, to resource task, t, as soon as possible. To this aim, x seeks to minimise the number of messages exchanged with potential providers required to release the required resource, r. The seeker, therefore, reveals all the information that, according to λ_x, the provider will require to release the resource in a single proposal. Since cost is incurred when information is revealed, however, this strategy will, at best, get the *baseline* utility; i.e. the utility expected if the provider indeed requires all information predicted to release the resource.

In the example introduced above, the seeker, x, will send $\lambda_x(y, r, t) = \{f_1, f_2, f_3, f_4\}$ to the provider in one message, and, if the request is successful, the utility gained will be:

$$\mu_x(y, r, t, \lambda_x(y, r, t)) = \Omega_x^t - (\Phi_y^r + Cost_x(\lambda_x(y, r, t), y))$$

This strategy ensures minimal messaging overhead if the seeker has accurate models of the policy and resource availability of providers.

3.2 Profit maximisation

The rationale for this strategy is to attempt to maximise the profit acquired in resourcing a task by attempting to reduce the information revelation costs in acquiring a resource. Using this strategy, the agent uses the models of other agents developed from past encounters to compute confidence values for each diagnostic information feature (i.e. their persuasive power). Suppose that the relative impact on a positive response from the provider in revealing features from $\lambda_x(y, r, t)$ are $f_3 > f_1$, $f_3 > f_2$, $f_1 > f_4$ and $f_2 > f_4$. Using this information, the agent will inform the potential provider of these features of the task context in successive messages according to this order when asked for justification of its request until agreement is reached (or the request fails).

In the above example, if the most persuasive justification (feature of the task context) succeeds, it will achieve an outcome of $\Omega_x^t - (\Phi_x^r + cost_x(f_3, y))$, if further justification is required either f_1 or f_2 is used, and so on.

Other strategies are, of course, possible. An immediate possibility is to order the features to be released on the basis of cost, or a combination of persuasive power and cost. Rather than discussing these relatively simple alternatives, in the following we discuss how such simple strategies could be combined.

3.3 Combined strategies

The rationale for these combined strategies is to capture the trade-off between presenting all the features of the task context in a single message, thereby, minimising communication, and attempting to extract as much utility as possible from the encounter (in this case by utilising information regarding relative persuasive power). One way of doing this, is to set a message threshold (a limit to the number of messages sent to a potential provider), σ_m. In other words, an agent can try to maximise utility (using the *profit maximising strategy*) in $\sigma_m - 1$ steps (or messages) and if the information revealed is insufficiently persuasive then the agent reveals all remaining task context features in the final message. It is easy to see that when σ_m is set to 1 then the agent adopts the *message minimisation* strategy, and if σ_m is set to $|\lambda_x(y, r, t)|$ this is equivalent to *profit maximisation*.

Condition	Description
RS	Random selection
SM	Simple memorisation of outcomes
SMMMS	SM + message minimising strategy
SMCS(0.5)	SM + combined strategy with $\sigma_c = 0.5$
SMCS(0.8)	SM + combined strategy with $\sigma_c = 0.8$
SMPMS	SM + profit maximising strategy
C4.5	Decision tree algorithm
kNN	k-Nearest neighbour- instance based algorithm
SC	Sequential covering- rule learning algorithm
SCMMS	SC + message minimising strategy
SCCS(0.5)	SC + combined strategy with $\sigma_c = 0.5$
SCCS(0.8)	SC + combined strategy with $\sigma_c = 0.8$
SCPMS	SC + profit maximising strategy

Figure 5: Experimental Conditions

Another way, is to identify the diagnostic features of the provider's decision (from the model), and compute the confidence values (persuasive power) for each feature. If the confidence value of a given feature exceeds some threshold, σ_c, then that feature is included in the set of information that will be revealed first (under the assumption that this set of features is most likely to persuade the provider to release the resource). If this does not succeed, the remaining features are revealed according to the profit maximisation strategy. For example, if f_3, f_2 and f_1 all exceed σ_c, these are sent in the first message, providing an outcome of $\Omega_x^t - (\Phi_y^r + Cost_x(\{f_1, f_2, f_3\}, y))$ if successful, and, if not, f_4 is used in a follow-up message.

Again, other strategies are possible such as computing a limited number of clusters of features on the basis of their persuasive power, or clustering by topic (if such background information is available). Our aim here is not to exhaustively list possible strategies, but to empirically evaluate the impact of utilising information from the models of others learned from past encounters to guide decisions regarding whom to engage in dialogue and what arguments to put forward to secure the provision of a resource (or, equivalently, a commitment to act). We turn to the evaluation of our model in the following section.

4. EVALUATION

In evaluating our approach, we implemented an agent society where a set of seeker agents interact with a set of provider agents with regard to resourcing their plans over a number of runs. Each provider is assigned a set of resources, and resources are associated with some charge, Φ_r. Providers also operate under a set of policy constraints that determine under what circumstances they are permitted to provide a resource to a seeker. The evaluation reported in this section is in two parts. In the first part, we demonstrate that it is possible to use evidence derived from argumentation to learn models of others' policies. To do this, we consider five experimental conditions in total (i.e. RS, SM, C4.5, kNN, and SC). These conditions are summarised in Figure 5.

The second part of this evaluation aims to demonstrate that a careful combination of machine learning and deci-

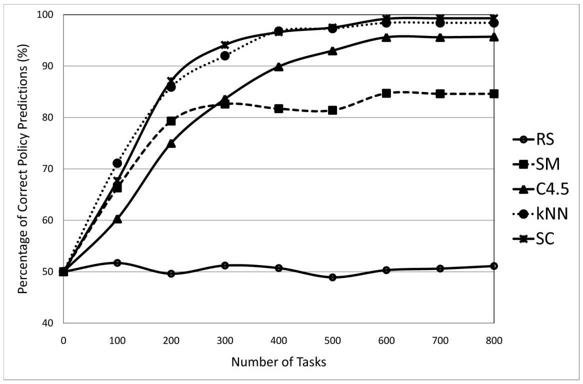

Figure 6: Policy prediction accuracy.

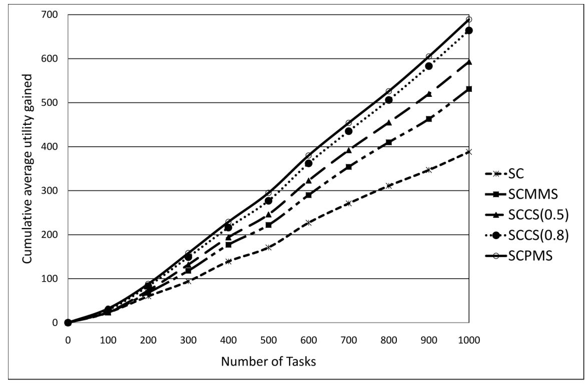

Figure 7: Cumulative average utility for SC

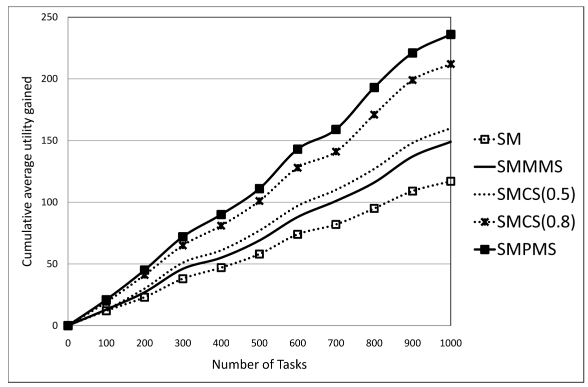

Figure 8: Cumulative average utility for SM

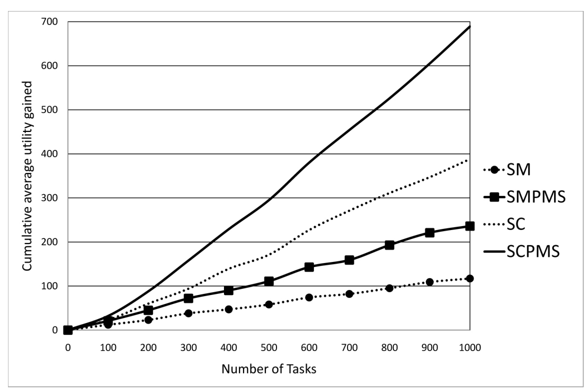

Figure 9: Cumulative average utility: SC vs. SM

sion theory can be used to aid agents in choosing who to partner with, and what information needs to be revealed in order to persuade the partner to release a resource. In this evaluation, we consider ten experimental conditions in total (i.e. SM, SMMMS, SMCS(0.5), SMCS(0.8), SMPMS, SC, SCMMS, SCCS(0.5), SCCS(0.8), SCPMS). Figure 5 describes the configurations tested in our experiments.

The scenario involves a team of five software agents (one seeker and four provider agents) collaborating to complete a joint activity over a period of three simulated days. There are five resource types, five locations, and five purposes that provide the possible task context of the use of a resource (375 possible task configurations). A task involves the seeker agent identifying resource needs for a plan and collaborating with the provider agents to see how that plan can be resourced. Experiments were conducted with seeker agents initialised with random models of the policies of provider agents. 100 runs were conducted in 10 rounds for each case, and tasks were randomly created during each run from the possible configurations. In the control condition, the seeker simply memorises outcomes from past interactions. Since there is no generalisation in the control condition, the *confidence* (or prediction accuracy) is 1.0 if there is an exact match in memory, else the probability is 0.5.

Figure 6 illustrates the performance of five algorithms we considered in predicting agents' policies through evidence derived from argumentation. The results show that sequential covering (SC), k-nearest neighbour (kNN), decision tree learner (C4.5) and simple memorisation (SM) consistently outperform the control condition (random selection, RS). Furthermore, both SC and kNN consistently outperform C4.5 and SM. It is interesting to see that, with relatively small training set, SM performed better than C4.5. This is, we believe, because the model built by C4.5 overfit the data. The decision tree was pruned after each set of 100 tasks and after 300 tasks the accuracy of the C4.5 model rose to about 83% to tie with SM and from then C4.5 performed better than SM. As we would expect, the average performance of the RS is in the region of 50%. Out of all the algorithms investigated here, SC was one of the best performers [1] and so we use it as the learning algorithm for the remaining parts of this evaluation. The SC algorithm also has the benefit of representing models of others' policies as rules, and hence are amenable to presentation to human decision makers.

Figure 7 compares the cumulative average utility of the seeker in five conditions, namely: SC, SCMMS, SCCS(0.5), SCCS(0.8), and SCPMS (see Figure 5). In each of these cases, rule learning (SC) is used to build models of others' policies. The results show that each of the five conditions evaluated here recorded increase in utility. However, SCMMS, SCCS(0.5), SCCS(0.8) and SCPMS significantly and consistently outperform SC. Although it does build a good policy model, this reduced performance is due to the absence of the decision-theoretic model for selecting y_{opt}. A similar comparison was done with five conditions using sim-

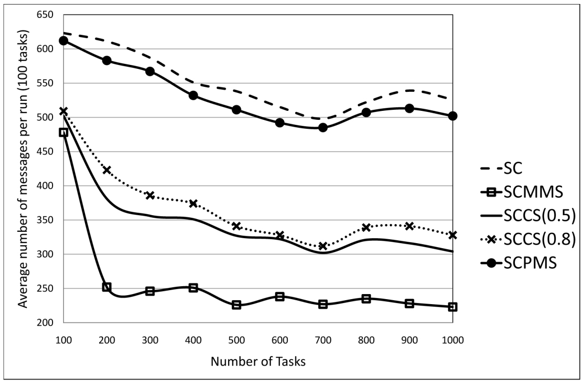

Figure 10: Average number of messages for SC

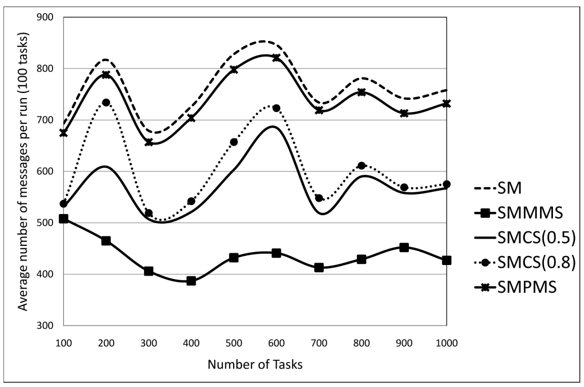

Figure 11: Average number of messages for SM

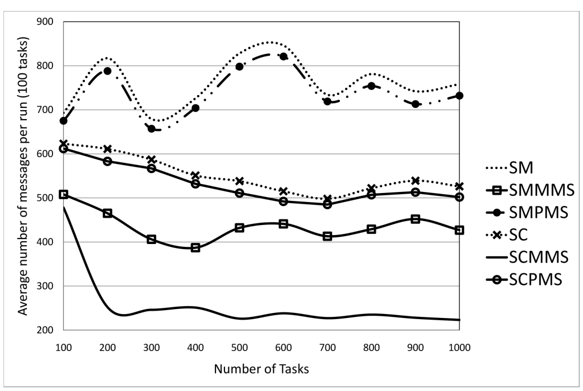

Figure 12: Average number of messages: SC vs. SM

ple memorisation (SM) and the results show similar patterns (see Figure 8). However, as shown in Figure 9, the utility the seeker gained in the SM configuration is small compared to that gained in scenarios where SC was used. Figure 9 compares the performance of agents that use SC and those that use SM. Results show that all configurations of SC (e.g. SCPMS, SC, etc) outperformed SM configurations throughout the experiment. This poor performance by SM stems from the fact that the seeker is unable to generalise from a number of examples; it only uses exact matches. The inability to build an accurate model of the policy of others reduces the effectiveness of the decision-theoretic model. Specifically, as shown in Figure 9, the lowest utility gained in the SC condition clearly outperformed the best result recorded in the SM configuration. This, further confirms our hypothesis that a combination of machine learning and decision theory will enable agents perform better than when there is no such combination.

In Figure 10 we plot the average number of messages exchanged in the five conditions against the number of tasks, where the seeker again uses rule learning (SC) to build policy models. Results show that, as expected, the number of messages exchanged in SCMMS condition was consistently and significantly lower than in the other four cases. For instance, just after 200 tasks, the communication overhead reduced to between 2 and 3 messages per task. The reason for this is simply because the seeker is (1) able to make an informed decision regarding which provider to approach for a given resource; and (2) able to preempt their information requirements and thereby present them without having to be asked. The SCMMS configuration uses a strategy that attempts to reduce the communication overhead by sending all the information it predicts will persuade the provider in one message. The SC condition (no argumentation strategy) has the highest average number of messages, similar to that for the profit maximising strategy, SCPMS. In the SCPMS case, the average number of messages is high because the seeker reveals minimal information in each message throughout the dialogue, leading to an increased number of messages, particularly if its policy models are accurate. A similar comparison was done with the five conditions using memorisation (SM), and the results show similar patterns in the number messages exchanged across the cases (see Figure 11). As shown in Figure 12, the number of messages in SM configurations is significantly greater that that in the corresponding SC case; the difference again being the beneficial effect of machine learning.

The combined strategy conditions with rule learning are worthy of particular note here. In SCCS(0.5) and SCCS(0.8), the seeker tries to find a compromise such that the communication is as low as possible while maximising profit. Both SCCS(0.5) and SCCS(0.8) require a similar average number of messages (Figure 10), but, referring back to Figure 7, SCCS(0.8) returns a cumulative average utility very close to the SCPMS case. The effect of this strategy is for the agent to reveal the information that is predicted to be most important to the provider in making a decision, while revealing other information features of the task context only when necessary for the negotiation to succeed. In this way, negotiation is focused on pertinent issues.

Tests of statistical significance were applied to the results of our evaluation, and they were found to be statistically significant by t-test with $p < 0.05$. Furthermore, for all the pairwise comparisons, the scenarios where machine learning was combined with decision theory consistently yielded higher utilities than those with simple memorisation. Similarly, scenarios where the decision-theoretic strategy mechanism was utilised constantly outperformed those without this mechanism. These results confirm our hypotheses; i.e. exploiting appropriate decision-theoretic and machine learning techniques can: (1) significantly improve the cumulative utility of dialogical outcomes; and (2) help to focus dialogue on pertinent issues for negotiation.

5. DISCUSSION

We started with the question "What do I need to say to convince you to do something?", and have presented and evaluated a model that starts to address this multi-faceted question. The approach combines argumentation, machine learning and decision theory to learn underlying social characteristics (e.g. policies/norms) of others and exploit the models learned to reduce communication overhead and improve strategic outcomes. We believe that this research contributes both to the understanding of argumentation strategy for dialogue among autonomous agents, and to applications of these techniques in agent support for human decision-making. In recent research, for example, Sycara et al. [9] report on a study into how software agents can effectively support human teams in complex collaborative planning activities. One area of support that was identified as important in this context is guidance in making policy-compliant decisions. This prior research focuses on giving guidance to humans regarding their own policies. An important and open question, however, is how can agents support humans in developing models of others' policies and using these in decision making? Our work seeks to bridge (part of) this gap. One of the limitations of the current research in this regard is due to the nature of the rules learned using sequential covering. Sequential covering is a greedy algorithm that does not necessarily find the best or smallest set of rules to cover the training instances, and further interpretation may be required if learned policies are to be presented to a human decision maker. Other techniques such as induction-based learning may help. In fact, Možina et al. [6] propose an induction-based machine learning mechanism, ABCN2, that uses argument structures to guide the process of inducing rules from examples; the arguments being inputs to the learning process. ABCN2 is an argument-based extension of CN2 rule learning, which out-performs CN2 in most tasks.

There are, of course, other aspects of our broad question that are not addressed here, which present interesting avenues for future research. In this paper we assume that the agent seeking to resource its plan makes a single decision per task about which provider to negotiate with; i.e. it has one go at resourcing a task. In reality, such a decision process should be iterative; i.e. if the most promising candidate fails to provide the resource, the next most promising is approached and the sunk cost incurred is taken into consideration, and so on. Furthermore, as indicated above, more sophisticated machine learning algorithms may be employed to build richer models of other agents, and hence further guide argumentation strategy. One possible avenue for future research in this regard is the use of background (or ontological) domain knowledge in machine learning. An agent could exploit knowledge of concept hierarchies in an ontology to better guide the learning of others' policies from specific instances; e.g. given positive examples of some agent providing a car and a van, we may assume the agent has no policy against providing ground vehicles. We believe that the research reported here, however, offers a solid basis from which to explore numerous issues of argumentation strategy.

6. CONCLUSIONS

In this paper, we have presented an agent decision-making mechanism where models of other agents are refined through evidence from past dialogues, and where these models are used to guide future argumentation strategy. Furthermore, we have empirically evaluated our approach and the results of our investigations show that decision-theoretic and machine learning techniques can individually and in combination significantly improve the cumulative utility of dialogical outcomes, and help to focus dialogue on pertinent issues for negotiation. We also argue that this combination of techniques can help in developing more robust and adaptive strategies for advising human decision makers on how a plan may be resourced (or a task delegated), who to talk to, and what arguments are most persuasive.

Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

7. REFERENCES

[1] C. D. Emele, T. J. Norman, F. Guerin, and S. Parsons. On the benefit of argumentation-derived evidence in learning policies. In *Proc. of the 3rd Intl. Workshop on Argumentation in Multi-Agent Systems*, Toronto, Canada, 2010.

[2] M. Finger, G. C. Bezerra, and D. R. Conde. Resource use pattern analysis for predicting resource availability in opportunistic grids. *Concurr. Comput. : Pract. Exper.*, 22(3):295–313, 2010.

[3] B. Grosz and S. Kraus. Collaborative plans for group activities. In *Proc. of the 13th Intl. Joint Conf. on Artificial Intelligence*, pages 367–373, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[4] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 12(2):315 – 334, 2002.

[5] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[6] M. Možina, J. Žabkar, and I. Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

[7] N. Oren, T. J. Norman, and A. Preece. Loose lips sink ships: A heuristic for argumentation. In *Proc. of the 3rd Int'l Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, pages 121–134, 2006.

[8] M. Sensoy, J. Zhang, P. Yolum, and R. Cohen. Context-aware service selection under deception. *Computational Intelligence*, 25(4):335 – 364, 2009.

[9] K. Sycara, T. J. Norman, J. A. Giampapa, M. J. Kollingbaum, C. Burnett, D. Masato, M. McCallum, and M. H. Strub. Agent support for policy-driven collaborative mission planning. *The Computer Journal*, 53(1):528–540, 2009.

[10] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.