Argumentation Strategies for Task Delegation*

Chukwuemeka David Emele[1], Timothy J. Norman[1], and Simon Parsons[2]

[1] University of Aberdeen, Aberdeen, AB24 3UE, UK
[2] Brooklyn College, City University of New York, 11210 NY, USA
{c.emele,t.j.norman}@abdn.ac.uk
parsons@sci.brooklyn.cuny.edu

Abstract. What argument(s) do I put forward in order to persuade another agent to do something for me? This is an important question for an autonomous agent collaborating with others to solve a problem. How effective were similar arguments in convincing similar agents in similar circumstances? What are the risks associated with putting certain arguments forward? Can agents exploit evidence derived from past dialogues to improve the outcome of delegation decisions? In this paper, we present an agent decision-making mechanism where models of other agents are refined through evidence derived from argumentation-based dialogues, and where these models are used to guide future argumentation strategy. We combine argumentation, machine learning and decision theory in a novel way that enables agents to autonomously reason about the constraints (e.g., policies or norms) that others are operating within, and make informed decisions about whom to delegate a task to. We demonstrate the utility of this novel combination of techniques through empirical evaluation in a plan resourcing domain. Our evaluation shows that a combination of decision-theoretic and machine learning techniques can significantly help to improve dialogical outcomes.

1 Introduction

Typically, 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, collaborators often engage in dialogue regarding the delegation of tasks, or sharing of resources. The formation of agreements may, however, be

* 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.

subject to policy restrictions. Such policies might regulate what resources may be released to an agent 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 [11].

One important aspect of collaborative activities is resource sharing and task delegation [4, 9]. 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 task delegation where agents operate under policies. 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 [13]. 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 agents that may be able to provide such a resource. The first question we address is how we may utilise evidence from past encounters to develop accurate models of others (see Section 2).

Given that agents are operating under policies, and some policies may prohibit an agent from performing an action under certain circumstances, how can we utilise models of others' policies that have been learned to devise a strategy for selecting an appropriate agent 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.[3] In the research presented in this paper, we intend to validate the following hypotheses: exploiting appropriate decision-theoretic and machine learning techniques can: (1) significantly improve the cumulative utility of dialogical outcomes; and (2) help to reduce communication overhead.

The remainder of this paper is organised as follows: Section 2 presents how to learn policies through evidence, and Section 3 discusses argumentation strategies. Section 4 reports the results of our evaluation, and Section 5 discusses related work and future direction. Section 6 concludes.

2 Learning policies

One of the core goals of this research is to learn models of the policies of others. In this section, we describe how the policies of others are captured and learned. We begin by formulating a mechanism to capture policies.

2.1 Policies

Agents in this framework have policies that govern how resources are deployed to others. In our model, policies are conditional entities (or rules) and so are relevant to an agent under specific circumstances only. These circumstances are characterised by a set of features such as the temperature of a room, the manufacturer of a car, the type of insurance cover, and so on.

[3] A preliminary version with initial results is reported in [3].

We define a feature as a characteristic of the prevailing circumstance within which an agent operates. Let $\mathcal{F}$ be the set of all features such that $f_1, f_2, \ldots \in \mathcal{F}$. 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 at a time, namely (i) *grant*, which means that the policy allows the agent to provide the resource when requested, and (ii) *deny*, which means that the policy prohibits the agent from providing the resource.

Definition 1. ***(Policies)*** *A policy is defined as a function $\Pi : 2^{\mathcal{F}} \rightarrow \{grant, deny\}$, which maps feature vectors of agents, to appropriate policy decisions.*

In order to illustrate the way policies may be captured in this model, we present the following examples.

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

$\mathbb{P}_2$**:** 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.

In the foregoing example, if *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 are no policies and/or availability constraints forbidding that. Furthermore, whenever a seeker's request is refused, the seeker may challenge the decision, and seek justifications for the refusal. This additional evidence is beneficial, and could be used to improve the model, hence, the quality of decisions made in future episodes. In the following section, we discuss argumentation-based negotiation as a mechanism that aids agents to gather additional evidence through dialogue.

2.2 Argumentation-based Negotiation

Negotiation may take many forms, but here, we focus on argumentation-based negotiation. We explore the evidence that argumentation-based dialogue provides in revealing underlying policy constraints, and thereafter we present the protocol employed in this research. Three important types of evidence are considered in this paper, namely: (i) seeking information about the issue under negotiation; (ii) providing explanations or justifications; and (iii) suggesting alternatives. This is not intended as an exhaustive list, but do represent three of the most common sources of evidence in argument-based dialogue in general.

Seeking further information When an agent receives a request to provide a resource, it checks whether or not it is permitted to honour the request. To do this, it must compare the details provided by the seeker with the policies it must operate within to make a decision. If the details of the task context provided by the seeker is insufficient for the provider to make a decision, it will need to seek further information. The seeker could use that information as input to try to model what policies the provider agent may be operating with. Such a request for further information could mean that there are specific values of certain features that may lead to different policy-governed decisions.

Suggesting alternatives. When an agent is unable to grant a request because there is either a policy restriction or a resource availability constraint, it may wish to suggest alternatives. For example, a seeker may request the use of a *helicopter* to transport relief materials in bad weather conditions. If the provider is prohibited from providing a *helicopter* in such conditions but permitted to provide a *jeep* then it may offer a *jeep* as an alternative for transporting those materials. If we assume that an agent will only suggest an alternative if that alternative is available and there is no policy that forbids its provision, then the suggestion provides evidence regarding the policies of the provider with respect to the suggested resource. While the issue of deception remains an open problem, some techniques for addressing this assumption have been investigated [12].

Justifications. Following a request for a resource, ultimately the provider agent will either agree to provide it or decline the request (though further information may be sought in the interim and suggestions made). In the case where the provider agent agrees to grant the request, the seeker agent obtains a positive example of a task context that the provider agent's policies permit the provision of the resource. On the other hand, if the request is refused then the seeker may seek further explanation for the refusal. The justification provided in response to the challenge may offer further evidence that may help to identify the underlying constraints.

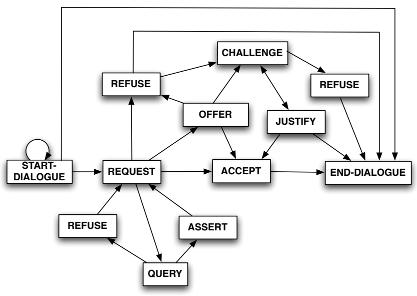

Fig. 1. The negotiation protocol.

Negotiation protocol Here, we present the protocol employed in this framework, which guides the negotiation process. Our approach is similar to [6] in negotiating for resources required to enact a plan. Figure 1 shows the negotiation protocol. To illustrate the sorts of interaction between agents, consider the example dialogue in Table 1. 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 (as in Table 1, *lines* 11 *to* 14), offer alternatives (*line* 10 in Table 1), or ask and receive more information about the attributes of requests (*lines* 4 *to* 9 in Table 1), then x can gather additional information regarding the policy rules guiding y concerning the provision of resources.

Table 1. Dialogue example.

#	**Dialogue Sequence**	**Locution Type**
1	x: Start dialogue.	START-DIALOGUE
2	y: Start dialogue.	START-DIALOGUE
3	x: Can I have a *helicopter* for $0.1M reward?	REQUEST
4	y: What do you need it for?	QUERY
5	x: To transport relief materials.	ASSERT
6	y: To where?	QUERY
7	x: A refugee camp near region XYZ.	ASSERT
8	y: Which date?	QUERY
9	x: On Friday 16/4/2010.	ASSERT
10	y: I can provide you with a *jeep* for $5,000.	OFFER
11	x: But I prefer a *helicopter*, why offer me a *jeep*?	CHALLENGE
12	y: I am not allowed to release a *helicopter* in volcanic eruption.	JUSTIFY
13	x: There is no volcanic eruption near region XYZ.	CHALLENGE
14	y: I agree, but the ash cloud is spreading, and weather report advises that it is not safe to fly on that day.	JUSTIFY
15	x: Ok then, I accept your offer of a *jeep*.	ACCEPT
16	y: That's alright. Good-bye.	END-DIALOGUE

Negotiation for resources takes place in a turn-taking fashion. The dialogue starts, and then agent x sends a request to agent y (e.g., *line* 3, Table 1). The provider, y, may respond by conceding to the request (accept), refusing, offering an alternative resource, or asking for more information (query) such as in *line* 4 in Table 1. If the provider agrees to provide the resource then the negotiation ends. If, however, the provider declines the request then the seeker may challenge that decision, and so on. If the provider suggests an alternative (*line* 10 in Table 1) then the seeker evaluates it to see whether it is acceptable or not. Furthermore, if the provider needs more information from the seeker in order to decide, the provider would ask questions that will reveal the features it requires to make a decision (query, assert/refuse in Figure 1). There is a cost attached to the revelation of private information to another agent. An agent might refuse to reveal a piece of information if doing so is expensive [10].

2.3 Learning from dialogue

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 features and policy decisions. 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 where $\vec{F}_y \in 2^{\mathcal{F}}$. Specifically, we investigate three classes of machine learning algorithms [7, 15], namely: decision tree learning (using C4.5), instance-based learning (using k-nearest neighbours), and rule-based learning (using sequential covering).

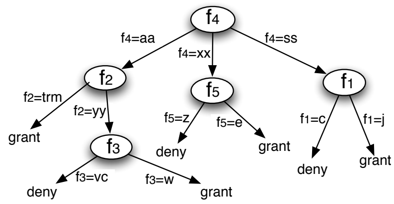

Fig. 2. Example decision tree.

Figure 2 shows an example tree representing an agent's policy model learned from interactions. Nodes of the decision tree capture features of an agent's policy, edges denote feature values, while the leaves are policy decisions.

3 Argumentation Strategies

We have described how the policies of other agents can be learned with the help of evidence derived from argumentation. In this section, we demonstrate the use of such structures in developing argumentation strategies for deciding which agent(s) to negotiate with and what arguments to put forward. Our model takes into account communication cost and utility 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. Agents play one of two roles: seeker and 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 such that $f_1, f_2, \ldots \in \mathcal{F}$. 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 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 operate according to their policies.

Each seeker agent $x \in \mathcal{A}$ has a function μ_x^r with signature $\mathcal{A} \times \mathcal{R} \times \mathcal{T} \times 2^{\mathcal{F}} \to \mathbb{R}$ that computes the utility gained if $x \in \mathcal{A}$ 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 $\mu_x^t : \mathcal{A} \times \mathcal{T} \times 2^{\mathcal{F}} \to \mathbb{R}$ 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.

Definition 2. ***(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 3. ***(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: $cost_x : \mathcal{F} \times \mathcal{A} \to \mathbb{R}$. 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 $Cost'_x : 2^{\mathcal{F}} \times \mathcal{A} \to \mathbb{R}$. 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. The actual destination may not be necessary, but would also be sufficient. The costs incurred in each case may differ, however. 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 about the context of the use of this resource is F.

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

Definition 4. ***(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 5. ***(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 providers 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 6. ***(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:

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

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 an accurate model of the policy and resource availability models 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, reducing the 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 reveal 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 the *profit maximising* strategy.

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 employed a simulated 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.

4.1 Experimental setup

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 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. 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.

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. 100 experimental runs were conducted. Each run consists of 8 rounds and at each round, 100 tasks were randomly generated (from the possible configurations) and assigned to a consumer, and the seeker attempts to delegate to others the provision of resources required to fulfill each task. The model built in each round is used to bootstrap the next round until the 8 rounds are completed. 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. We consider four experimental conditions in total, summarised in Table 2 (i.e. SM, C4.5, kNN, and SC). In SM, simple memorisation of outcomes is used. In C4.5, C4.5 decision tree classifier [15] is used. In kNN, k-nearest neighbour algorithm [1] is used. In SC, sequential covering rule learning algorithm [5] is used.

Table 2. Experimental Conditions

Condition	**Description**
SM	Simple memorisation of outcomes
SMMMS	SM + message minimising strategy
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
SCPMS	SC + profit maximising strategy

The second part of this evaluation aims to demonstrate that a careful combination of machine learning and decision 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 the resource. In this evaluation, we consider six experimental conditions in total (i.e. SM, SMMMS, SMPMS, SC, SCMMS, SCPMS). Table 2 outlines the configurations tested in our experiments.

4.2 Results

Figure 3 illustrates the performance of five algorithms we considered in predicting agents' policies through evidence derived from argumentation. The results

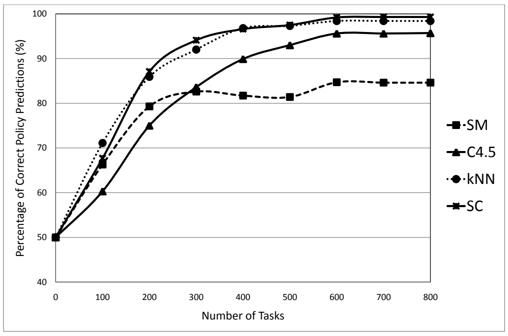

Fig. 3. Prediction accuracy of different cases.

show that sequential covering (SC) and k-nearest neighbour (kNN) consistently outperform the control condition (simple memorisation, SM). Furthermore, both SC and kNN consistently outperform C4.5. 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. Out of all the algorithms investigated here, SC was one of the best performers [2] 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.

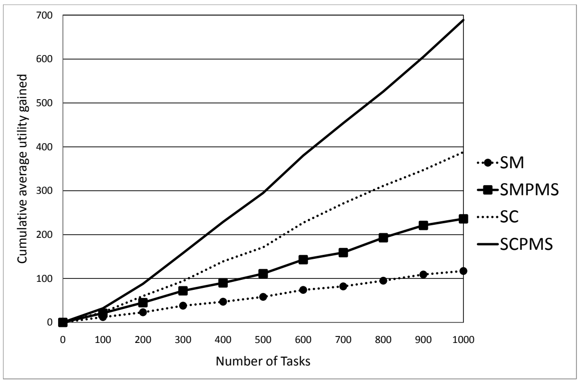

Fig. 4. SC vs. SM, cumulative average utility

As shown in Figure 4, the utility the seeker gained in the SM configuration is small compared to that gained in scenarios where SC was used. Figure 4 compares the performance of agents that use SC and those that use SM. Results show that all configurations of SC (i.e. SCPMS, SC) 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 4, 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.

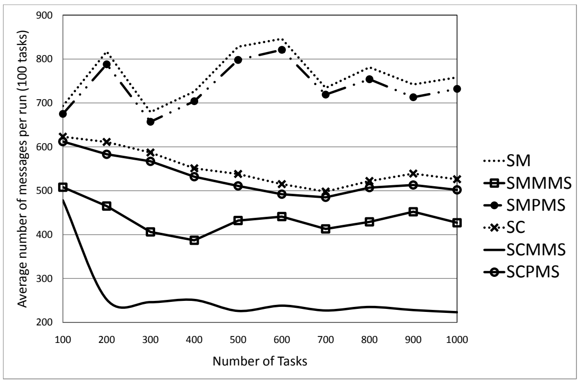

Fig. 5. SC vs. SM, number of messages exchanged

Figure 5 gives a graphical illustration of the average number of messages exchanged in the four conditions. The number of messages the seeker exchanged in the SM configuration is more than the corresponding case in SC configurations. The reason for this, we believe, is because the *seeker* in the SM setup is unable to build a reliable model of the provider agent's policies and resource availabilities, and will struggle to preempt the information requirements of partners. Thus, the results show that the average number of messages per task varies between 4 and 5 in SMMMS, and between 6 and 9 in both SM and SMPMS.

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 decision theory was utilised constantly outperformed those without decision theory. These results show that the combination of techniques can help increase the performance of agents. These results confirm our hypotheses, which state that exploiting appropriate decision-theoretic and machine learning techniques can: (1) significantly improve dialogical outcomes; and (2) help to reduce communication overhead.

5 Discussion

The results we have presented show that a decision-making mechanism based on a combination of decision-theoretic and machine learning techniques can clearly help agents to improve the cumulative utility of dialogical outcomes. Our approach represents the first model for combining 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. There is, however, some prior research in combining machine learning and argumentation, and in using argument structures for machine learning. In that research, Možina et al. [8] propose an induction-based machine learning mechanism using argumentation. The work implemented an argument-based extension of CN2 rule learning (ABCN2) and showed that ABCN2 out-performed CN2 in most tasks. However, the framework developed in that research will struggle to learn and build an accurate model of policies from argumentation-derived evidence, which is the main issue we are addressing in our work. Also, the authors assume that the agent knows and has access to the arguments required to improve the prediction accuracy, but we argue that it is not always the case. As a result, we use dialogue to tease out evidence that could be used to improve performance.

In recent research, Sycara et al. [14] investigate agent support for human teams in which software agents aid the decision making of team members during collaborative planning. 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? We use a novel combination of techniques to build accurate models of others' policies, and use these to aid decision making. We believe that our 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 the evaluation presented in this paper, we assume that the seeker makes a single decision per task about which provider to choose, irrespective of whether it fails or succeeds. In our future work, we plan to make the decision process more iterative such that if the most promising candidate fails to provide the resource, the next most promising is approached and the sunk cost incurred while interacting with the previous provider is taken into account in computing the total cost of resourcing the task, etc. We are hoping that some of these ideas will provide helpful feedback to future research on developing strategies for delegation in which there might be a cost for failing to resource a task.

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 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 reduce communication overhead. The results also demonstrate 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.

References

1. Duda, R., Hart, P.: Pattern classification and scene analysis. John Wiley & Sons (1973)
2. Emele, C.D., Norman, T.J., Guerin, F., Parsons, S.: 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)
3. Emele, C.D., Norman, T.J., Parsons, S.: Argumentation strategies for plan resourcing. In: Yolum, Tumer, Stone, Sonenberg (eds.) Proc. of 10th Int'l. Conf. on Autonomous Agents and Multiagent Systems. pp. 913–920. Taipei, Taiwan (2011)
4. Grosz, B., Kraus, S.: Collaborative plans for group activities. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. pp. 367–373 (1993)
5. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan Kaufmann, 2nd edn. (2006)
6. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. Journal of Logic, Language and Information 12(2), 315 – 334 (2002)
7. Mitchell, T.M.: Machine Learning. McGraw Hill (1997)
8. Možina, M., Žabkar, J., Bratko, I.: Argument based machine learning. Art. Intel. 171(10-15), 922–937 (2007)
9. Norman, T.J., Reed, C.: A logic of delegation. Art. Intel. 174(1), 51–71 (2010)
10. Oren, N., Norman, T.J., Preece, A.: Loose lips sink ships: A heuristic for argumentation. In: Proc. of the 3rd Int'l Workshop on ArgMAS. pp. 121–134 (2006)
11. Sensoy, M., Norman, T.J., Vasconcelos, W.W., Sycara, K.: OWL-POLAR: Semantic policies for agent reasoning. In: Proc. of the 9th International Semantic Web Conference. LNCS, vol. 6414, pp. 679–695. Springer-Verlag (2010)
12. Sensoy, M., Zhang, J., Yolum, P., Cohen, R.: Context-aware service selection under deception. Computational Intelligence 25(4), 335–364 (2009)
13. Sukthankar, G., Sycara, K.: Analyzing team decision-making in tactical scenarios. The Computer Journal 53(5), 503–512 (2010)
14. Sycara, K., Norman, T.J., Giampapa, J.A., Kollingbaum, M.J., Burnett, C., Masato, D., McCallum, M., Strub, M.H.: Agent support for policy-driven collaborative mission planning. The Computer Journal 53(5), 528–540 (2010)
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2nd edn. (2005)