# 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 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 reason about constraints (e.g., policies) that others are operating within, and make informed decisions about whom to delegate a task to. We demonstrate the utility of this novel approach 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

It is typical in collaborative settings for agents (human or artificial) to work together, act on each others' behalf, share resources, etc [4,8]. This presupposes that there exist some kind of relationship or agreement between collaborators. Regardless of whether such relationships are transient or permanent, collaborators often engage in dialogue regarding task delegation, or resources sharing. Agents in such settings may, however, be 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.

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 an approach based on decision theory, 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. We explore, in this paper, strategies for task delegation where agents operate under policies, and we intend to validate the following hypothesis: *agents that build more accurate models of others and use this to drive argumentation strategy will perform better*

M. Cossentino et al. (Eds.): EUMAS 2011, LNAI 7541, pp. 80–96, 2012.

*than those that do not. More specifically, exploiting appropriate decision-theoretic and machine learning techniques will mean that agents can: (1) significantly improve the cumulative utility of dialogical outcomes; (2) reduce communication overhead; and (3) strike a balance between maximising utility and minimising communication overhead.*

The remainder of this paper is organised as follows: Section 2 presents our framework and describes how policies are learned through evidence, and Section 3 discusses our decision-theoretic model. Section 4 presents a number of strategies for selecting arguments. Section 5 reports the results of our evaluation, and Section 6 discusses related work and future direction. Section 7 concludes.

## 2 Our Framework

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

### 2.1 Agents' Policies

Agents' policies regulate how tasks are delegated or resources deployed to others. Here, we develop an abstract model of policies, which provides the basis for designing a framework that allows agents to reason about others' policies as they collaborate, communicate and coordinate their activities. In this model, we assume that agent policies can be described in terms of features that characterise the prevailing circumstances. Our approach of using features to model systems has been used in data mining and machine learning problems [2,10], where features capture the attributes and characteristics of objects. We model policies as conditional entities that are relevant to an agent under specific circumstances only. These circumstances are characterised by a set of features such as the type of resource required, the location of an operation, and so on.

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) *decline*, which means that the policy prohibits the agent from providing the resource.

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

We illustrate, by examples, the way policies may be captured in this model.

$\mathbb{P}_1$**:** You are **prohibited** from releasing a $helicopter$ to any agent if the weather report says there are volcanic clouds (vc) in the location the agent intends to deploy the $helicopter$.

$\mathbb{P}_2$**:** You are **permitted** to release a $helicopter$ $(h)$, to any agent if the $helicopter$ is required for transporting relief materials $(trm)$ and the weather is good.

$\mathbb{P}_3$**:** You are **permitted** to release a $jeep$ $(j)$ to any agent for any purpose, irrespective of the day and the weather report.

In the above example, if a *helicopter* is intended to be deployed in an area with volcanic clouds then the provider is prohibited from providing the resource but might offer a ground vehicle (e.g., *jeep*) to the consumer if the resource is available. Policies are important factors that regulate agents' behaviour in a society. Given that policies are often private, and agents are required to work together as they collaborate to solve a problem then *how can agents identify what policies others are working within*? Our claim is that there is useful evidence that one can extract from interactions with other agents. Such additional evidence can help to build more accurate models of others' policies. In the next section, we discuss how argumentation-based dialogue allows us to gather such useful evidence.

### 2.2 Argumentation-Derived Evidence (ADE)

We explore the evidence that argumentation-based dialogue provides in revealing underlying policy constraints, and thereafter we present the interaction 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 [14].

**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 consumer with the policies it must operate within to make a decision. If the details of the task context provided by the consumer is insufficient for the provider to make a decision, it will need to seek further information. The consumer 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 consumer 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 consumer agent obtains a positive example of a task context that

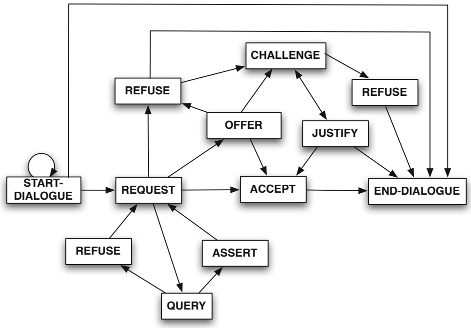


**Fig. 1.** The interaction protocol

the provider agent's policies permit the provision of the resource. On the other hand, if the request is refused then the consumer 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.

**Interaction Protocol.** Here, we present the protocol employed in this framework which guides the interaction between agents (see Figure 1). Our approach is similar to [1,5] in negotiating for resources required to enact a plan. To illustrate the sorts of interaction between agents, consider the example dialogue in Table 1. Let $x$ and $y$ be consumer 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.

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 consumer may challenge that decision, and so on. If the provider suggests an alternative (*line* 10 in Table 1) then the consumer evaluates it to see whether it is acceptable or not. Furthermore, if the provider needs more information from the consumer 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 [9].

### 2.3 Policy Modelling through Argumentation-Derived Evidence (ADE)

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

**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 |

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 $(\vec{F}_y, decline)$ 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 [6,15], namely: decision tree learning (using C4.5), instance-based learning (using k-nearest neighbours), and rule-based learning (using sequential covering).

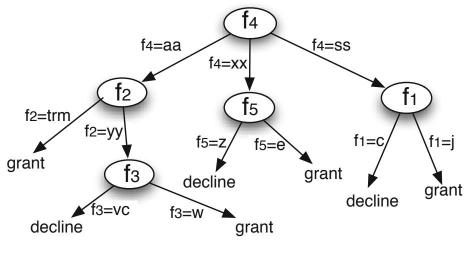


**Fig. 2.** Example decision tree

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 Decision-Theoretic Model

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 5.

Let $\mathcal{A}$ be a society of agents. Agents play one of two roles: consumer 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 consumer 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 consumer agent $x \in \mathcal{A}$ has a function $\mu_x^r$ with signature $\mathcal{A} \times \mathcal{R} \times \mathcal{T} \times 2^{\mathcal{F}} \rightarrow \mathbb{R}$ that computes the utility gained if $x \in \mathcal{A}$ 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 $\mu_x^t : \mathcal{A} \times \mathcal{T} \times 2^{\mathcal{F}} \rightarrow \mathbb{R}$ 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 consumer 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 consumer 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:*

$$\mu_x(y, r, t, F) = \Omega_x^t - (\Phi_y^r + Cost_x(F, y)) \tag{1}$$

*where $\Omega_x^t$ is the reward received by $x$ for resourcing task $t$, $\Phi_y^r$ 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[1]. Even in a cooperative setting, there is a chance that information revealed to others can be exploited. We agree, however, that there might be situations where revealing more information could lead to better outcomes for both agents. Notwithstanding, this does not rule out the fact that revealing too much information in a current dialogue might damage an agent's chance of winning a future argument. This line of argument has seen use in many practical applications. For example, during job interviews, applicants often plan to reveal information that they think is likely to project them as the right candidate for the job. In that regard, they usually plan not to present any information that could reveal otherwise. In another practical example, [11] speculates that certain government spying organisations are easily able to break most forms of encryption. However, when required to present evidence in a court of law, these organisations try to pose arguments that avoid revealing such information. This is because they consider revealing such information to be expensive.

**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: $cost_x : \mathcal{F} \times \mathcal{A} \rightarrow \mathbb{R}$. 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 $f \in F$.*

$$Cost_x(F, y) = \sum_{f \in F} cost_x(f, y) \tag{2}$$

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. 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.

[1] It is worth noting that the utility derived by the agent to whom the information was revealed is beyond the scope of this paper, and so is not discussed here.

**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 $y$:*

$$Pr(Yes|y, r, F) = Pr(Permitted|y, r, F) \times Pr(Avail|y, r) \tag{3}$$

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:*

$$E(x, y, r, t, F) = \mu_x(y, r, t, F) \times Pr(Yes|y, r, F) \tag{4}$$

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 $\lambda_x : \mathcal{A} \times \mathcal{R} \times \mathcal{T} \to 2^{\mathcal{F}}$*

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

$$y_{opt} = \arg\max_{y \in \mathcal{A}} E(x, y, r, t, F) \text{ s.t. } F = \lambda(y, r, t) \tag{5}$$

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 end, we present strategies for selecting arguments in the next section.

## 4 Strategies for Selecting Arguments

We 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 consumer) 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.

### 4.1 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. To effectively specify this heuristic, we define a function $power$, which returns the strength or persuasive power of a feature in leading to a positive response from the provider as follows.

**Definition 7.** ***(Persuasive Power)*** *The persuasive power of a single item of information $f \in \mathcal{F}$ with respect to a specific agent, $y \in \mathcal{A}$ regarding a specific resource $r \in \mathcal{R}$, denoted as $power_x(y, r, f)$, is defined as the confidence that agent $x$, has in the fact that revealing $f$ to another agent, $y$, will contribute positively towards persuading $y$ along the desired (or positive) direction regarding the provision of $r$.*

$$power_x : \mathcal{A} \times \mathcal{R} \times \mathcal{F} \to \mathbb{R} \tag{6}$$

Computationally, this can be done by generating confidence values (or probabilities over the information features) with respect to how features have contributed to positive responses in past dialogues. We leave the choice of which approach to use in generating the persuasive power or strength of arguments (that is, information feature) to system designers who can decide based on their objectives and the peculiarities of the domain of interest.

Using this strategy, the agent uses the models of other agents developed from past encounters to compute confidence values (or persuasive power) for each diagnostic information feature. Suppose that the persuasive power of the 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 $\Omega_x^t - (\Phi_y^r + Cost_x(f_3, y))$, 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.

### 4.2 Message Minimisation

The rationale for the use of this first strategy is for the consumer 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 consumer, therefore, reveals all the information that, according to $\lambda_x$, 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 consumer, $x$, 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 consumer has an accurate model of the policy and resource availability models of providers. A number of variations can be formulated for this strategy. An immediate possibility is for agents to anticipate the next question of the other party and provide a response in advance. To demonstrate this variation, we refer to the dialogue captured earlier in Table 1. In *line 4* of the dialogue, $y$ asks *"What do you need it for?"*. Agent $x$, rather than just respond with *"To transport relief materials"*., may prefer to respond with *"To transport relief materials to refugee camp near region XYZ"*. In our evaluation, we implement the variation of message minimisation strategy that reveals all the information (in one response) that, according to $\lambda_x$, will be required to release $r$.

### 4.3 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), $\sigma_m$. 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 $\sigma_m$ is set to 1 then the agent adopts the *message minimisation* strategy, and if $\sigma_m$ 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, $\sigma_c$, 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 $\sigma_c$, 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.

# 5 Evaluation

In evaluating our approach, we employed a simulated agent society where a set of consumer agents interact with a set of provider agents with regard to resourcing tasks assigned to them, and we test the following hypothesis:

– *Hypothesis 1:* Incremental revelation of information ordered by *persuasive power* significantly improves the cumulative utility of dialogical outcomes.
– *Hypothesis 2:* Anticipation of information needs of others leads to significant reduction in the number of messages exchanged (i.e. communication overhead).

## 5.1 Experimental Setup

The scenario involves a team of five software agents (one consumer 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 consumer agent identifying resource needs for a plan and collaborating with provider agents to see how that plan can be resourced. Experiments were conducted with consumer agents initialised with random models of the policies of provider agents. In the control condition, the consumer simply memorises outcomes from past interactions (see SM configuration below). 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, otherwise we assume there is a 50:50 chance of the prediction being accurate. Typically, if there is no exact match the control condition does not look for the best match (because this will involve some generalisation, which is not allowed in this configuration). In other configurations involving machine learning (e.g., SC, see below), the confidence can be generalised from past interactions.

In our experimental setup, the consumer's policies allow it to delegate the provision of resources to any of the four providers in the system. However, the cost of revealing information to various providers differ. This could be used to model such things as trust (or distrust). Each provider is assigned a set of resources, and resources are associated with some charge, $\Phi_r$. Providers also operate under a set of policy constraints that determine under what circumstances they are permitted to provide a resource to a consumer. We conducted 800 experiments, and in each experiment 100 tasks were randomly generated (from the 375 possible task configurations) and assigned to a consumer, and the consumer attempts to delegate to others the provision of resources required to fulfil each task. Based on previous experience (that is, the policy model built so far), an agent tries to predict whether or not a provider is permitted to release a given resource. The prediction is logged and after the experiment the predictions are checked against the agent's policies. The more accurate the policy model built, the more accurate the predictions.

For example, suppose a consumer is assigned to deliver relief materials to victims of a natural disaster in a certain location. The consumer then probes the environment for provider agents that can provide required resources. After identifying potential providers, it employs our decision model to select the most promising candidate. Thereafter, it engages in argumentation-based negotiation with the agent in an attempt to acquire such resources. The procedure for the negotiation follows the interaction protocol (presented in Section 2). For example, in the argumentation-based dialogue captured in Table 1, the consumer $x$, initiated the dialogue. The provider had a policy that forbid it from releasing a *helicopter* but was allowed to release a *jeep* and so it offered the consumer a *jeep*.

**Table 2.** Experimental Conditions

| **Condition** | **Description** |
|---|---|
| SM | Simple memorisation of outcomes |
| SMMMS | SM + message minimising strategy |
| SMPMS | SM + profit maximising strategy |
| SC | Sequential covering rule learning algorithm |
| SCMMS | SC + message minimising strategy |
| SCPMS | SC + profit maximising strategy |

In this evaluation, we aim 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. We consider six experimental conditions in total (i.e. SM, SMMMS, SMPMS, SC, SCMMS, SCPMS). In an earlier research [3], Emele *et al.* explored the performance of different classes of machine learning techniques in building accurate models of the policies of others through argumentation-derived evidence. Out of all the algorithms investigated in that research, SC was one of the best performers, 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.

Table 2 outlines the configurations tested in our experiments while Table 3 captures the cost that the consumer associates with revealing the various features of a task to different providers ($y_1, y_2, y_3$ and $y_4$). We assume that these costs are constant throughout the experiment. This simplification is to enable us concentrate on such things as identifying the most promising candidate and what information features are more persuasive in a given context. In addition, the reward offered for fulfilling tasks were randomly generated and range between \$18 and \$25. The price of resources were also randomly generated and lies between \$7 and \$12. Once the price of resources are generated at the beginning of the experiment, it remains the same throughout the experiment. Again, the reason for this is both to simplify the experiment and to allow us investigate the effect of our decision model without bias.

**Table 3.** Cost associated with revealing various features to various providers

| **Feature** | $Cost_x(F, y_1)$ | $Cost_x(F, y_2)$ | $Cost_x(F, y_3)$ | $Cost_x(F, y_4)$ |
|---|---|---|---|---|
| $Resource$ | \$2 | \$2 | \$2 | \$2 |
| $Day$ | \$1 | \$1 | \$1 | \$2.50 |
| $Location$ | \$4 | \$1.50 | \$1 | \$3 |
| $Purpose$ | \$6 | \$4 | \$7 | \$2 |

### 5.2 Results

We aim to confirm whether or not agents that utilise a combination of machine learning and decision theory to guide their argumentation strategies can perform better than those that do not.

**Hypothesis 1**
In a set of experiments, we evaluate the performance of incremental revelation of information features, ordered by *persuasive power* of arguments, on the cumulative utility gained from dialogical encounters.

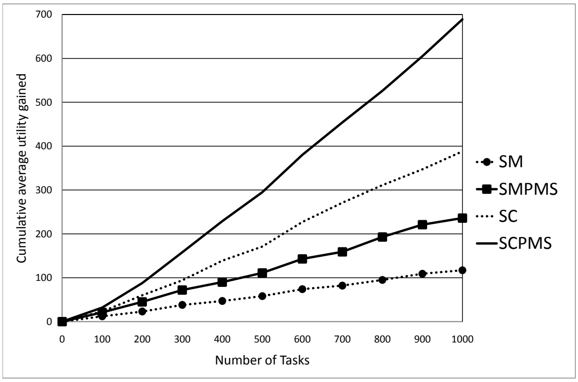


**Fig. 3.** SC vs. SM, cumulative average utility

Figure 3 t agents perform better (that is, gain higher cumulative average utility) when they reveal information in incremental fashion given that such information is ordered by its *persuasive power*. Regardless of how simple or sophisticated an agents learning approach is, results show that if agents can (somehow) order the information according to its *persuasive power* and incrementally reveal them during negotiation the agent is more likely to perform better and thereby gain higher cumulative average utility. More specifically, the SMPMS consistently gained higher utility than the SM configuration. Likewise, the SCPMS recorded greater improvement in the cumulative utility gained than its counterpart (i.e. SC). For example, after 800 tasks, the cumulative average utility gained by agents using the incremental revelation approach had risen above \$190 and \$520 (in SMPMS and SCPMS configurations respectively) while the configuration that reveals information without considering the *persuasive power* approaches \$96 and \$312 (in SM and SC configurations respectively). Clearly, the configurations in which incremental revelation of information takes into account the persuasive power significantly and consistently outperforms those without such consideration. These results show that incremental revelation of information ordered by *persuasive power* significantly improves the cumulative utility of dialogical outcomes, which confirms our hypothesis.

In order to test the statistical significance of the results of our evaluation, we carried out a paired t-test to determine whether or not the null hypothesis $H_0$ should be rejected in favour of the alternative hypothesis $H_1$.

- $H_0$ = There is NO significant difference in the performance of agents that revealed information incrementally (based on the *persuasive power*) and those that revealed information without paying attention to the *persuasive power* of arguments being put forward.
- $H_1$ = There is a significant difference between the performance of agents that revealed information incrementally (based on the *persuasive power*) and those that revealed information without paying attention to the *persuasive power* of arguments being put forward.

**Table 4.** Statistical analysis of utility gained across different configurations

| **Configuration** | **$t$-statistic** | **p-value** | **95% Conf. interval** | | **Significant** |
|---|---|---|---|---|---|
| | | | From | To | |
| SM vs SMPMS | 1.71 | 0.033 | 0.56 | 2.86 | Yes |
| SC vs SCPMS | 24.34 | $\ll 0.001$ | 23.19 | 25.49 | Yes |

In Table 4, we summarise the results of the statistical analysis performed on the experimental data in various configurations of the agent. From the statistical analysis, the results show that there is a significant difference in the performance of agents in the SM vs SMPMS and SC vs SCPMS configurations. With each pairwise comparison recording $p < 0.05$, we reject the null hypothesis, and conclude that agents that reveal information incrementally (based on the persuasive power) perform better than those that reveal information without considering the persuasive power of arguments being put forward. This further confirms our hypothesis.

**Hypothesis 2**

In a series of experiments, we evaluate the effectiveness of anticipating the information needs of others, and how it affects the number of messages exchanged (i.e. communication overhead) during dialogical encounters. We considered the following configurations — SM, SMMMS, SC, and SCMMS.

For all the configurations considered, the number of messages exchanged during dialogical encounters was considerably reduced in configurations where agents anticipate the information needs of others (and therefore provide it ahead of time) than those without such capability. Figures 4 shows the effectiveness of anticipation of information needs of others using simple memorisation, and rule learning respectively. In each case, results clearly show that communication overhead is reduced when agents anticipate others' information needs. Irrespective of the complexity or simplicity of the learning approach employed, results show that if agents can accurately predict the information requirement of other partners in collaborative problem solving activities then they can significantly reduce the communication overhead. For example, after 600 tasks, the number of messages exchanged per 100 tasks by agents that anticipate the information needs of others had fallen below 595 and 240 messages (averaging about 6, 3 messages per task) in SMMMS and SCMMS configurations respectively, while configurations in which agents do not anticipate others' information needs is above 845 and 510 messages

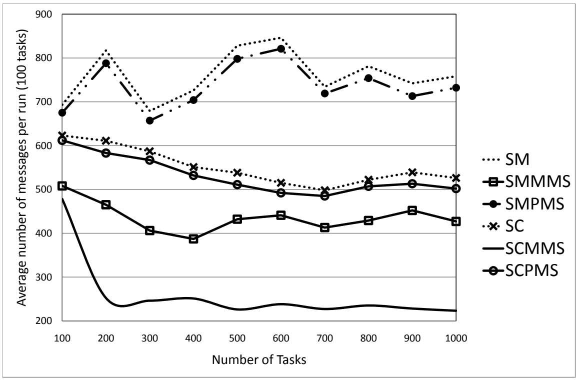


**Fig. 4.** SC vs. SM, number of messages exchanged

per 100 tasks (that is, more than $8$ and $5$ messages per task) in SM, and SC configurations respectively. Clearly, these results show that anticipation of information needs of others consistently and significantly reduces the number of messages exchanged (i.e. communication overhead) during dialogical encounters. This confirms our hypothesis.

We carried out a paired t-test analysis to test the statistical significance of the results of our evaluation. The null hypothesis $H_0$ and the alternative hypothesis $H_1$ are as follows:

- $H_0$ = The number of messages exchanged by agents that anticipate the information needs of others is equal to those exchanged by agents that do not anticipate the information needs of others.
- $H_1$ = There is a significant difference between the number of messages exchanged by agents that anticipate the information needs of others and those that do not anticipate others' information needs.

**Table 5.** Statistical analysis of messages exchanged across configurations

| **Configuration** | **$t$-statistic** | **p-value** | **95% Conf. interval** | | **Significant** |
|---|---|---|---|---|---|
| | | | From | To | |
| SM vs SMMMS | 1.90 | 0.030 | 0.75 | 3.05 | Yes |
| SC vs SCMMS | 14.32 | $\ll 0.001$ | 13.17 | 15.47 | Yes |

In Table 5, we summarise the results of the statistical analysis performed on the experimental data in various configurations of the agent. From the statistical analysis, the results show that there is a significant difference in the number of messages exchanged by agents in the SM vs SMMMS and SC vs SCMMS configurations. With each pairwise comparison recording $p < 0.05$, we reject the null hypothesis, and conclude that agents that anticipate the information needs of others perform better (lead to significant reduction in communication overhead) than those that do not consider such anticipation in their interactions. This further confirms our hypothesis.

## 6 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 their performance both in terms of utility and communication overhead. 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. [7] propose an induction-based machine learning mechanism using argumentation. 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. [13] 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 consumer 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.

## 7 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. Amgoud, L., Parsons, S., Maudet, N.: Argument, dialogues and negotiation. In: Horn, W. (ed.) Proceedings of European Conference on Artificial Intelligence 2000, pp. 338–342. IOS Press, Amsterdam (2000)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2), 245–271 (1997)
3. Emele, C.D., Norman, T.J., Guerin, F., Parsons, S.: On the Benefits of Argumentation-Derived Evidence in Learning Policies. In: McBurney, P., Rahwan, I., Parsons, S. (eds.) ArgMAS 2010. LNCS, vol. 6614, pp. 86–104. Springer, Heidelberg (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. 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)
6. Mitchell, T.M.: Machine Learning. McGraw Hill (1997)
7. Možina, M., Žabkar, J., Bratko, I.: Argument based machine learning. Art. Intel. 171(10-15), 922–937 (2007)
8. Norman, T.J., Reed, C.: A logic of delegation. Art. Intel. 174(1), 51–71 (2010)
9. 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)
10. Ren, L., Shakhnarovich, G., Hodgins, J.K., Pfister, H., Viola, P.: Learning silhouette features for control of human motion. ACM Trans. Graph. 24(4), 1303–1331 (2005)
11. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., New York (1993)
12. Sensoy, M., Zhang, J., Yolum, P., Cohen, R.: Context-aware service selection under deception. Computational Intelligence 25(4), 335–364 (2009)
13. 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)
14. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. SUNY Press, USA (1995)
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann (2005)