Loose Lips Sink Ships*: a Heuristic for Argumentation

Nir Oren, Timothy J. Norman, and Alun Preece

Department of Computing Science, University of Aberdeen, Aberdeen, AB24 3UE, Scotland
noren,tnorman,apreece@csd.abdn.ac.uk

Abstract. While researchers have looked at many aspects of argumentation, an area often neglected is that of argumentation strategies. That is, given multiple possible arguments that an agent can put forth, which should be selected in what circumstances. In this paper, we propose a heuristic that implements one such strategy, namely revealing as little information as possible to other dialogue participants. After formalising the concept and presenting a simple argumentation framework in which it can be used, we show a sample dialogue utilising the heuristic. We conclude by exploring ways in which this heuristic can be employed and a discussion of future work is made which will allow for the use of our approach in more complicated, realistic dialogues.

1 Introduction

Argumentation has emerged as a powerful reasoning mechanism in many domains. One common dialogue goal is to persuade, where one or more participants attempt to convince the others of their point of view. This type of dialogue can be found in many areas including distributed planning and conflict resolution, education and in models of legal argument.

At the same time that the breadth of applications of argumentation has expanded, so has the sophistication of formal models designed to capture the characteristics of the domain. In particular, Prakken [1] has focused on legal argumentation, and has identified four layers with which such an argumentation framework must concern itself. These are:

- The *logical layer*, which allows for the representation of basic concepts such as facts about the world. Most commonly, this layer consists of some form of non–monotonic logic.
- The *dialectic layer*, in which argument specific concepts such as the ability of an argument to defeat another are represented.
- The *procedural layer* governs the way in which argument takes place. Commonly, a dialogue game [2] is used to allow agents to interact with each other.

* This was a motto used in World War II to remind people not to inadvertently reveal possibly secret information.

– The *heuristic layer* contains the remaining parts of the system. Depending on the underlying layers, these may include methods for deciding which arguments to put forth and techniques for adjudicating arguments.

While many researchers have focused on the lowest two levels (excellent surveys can be found in [3, 1, 4]), and investigation into various aspects of the procedural layer is ongoing (for example, [5, 6]), many open questions remain at the heuristic level.

In this paper, we propose a decision heuristic for an agent allowing it to decide which argument to put forth. The basis for our idea is very simple; an agent should, while attempting to win a dispute, reveal as little of what it knows as possible. This heuristic has seen use in many real world situations. For example, it has long been speculated [7] 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 first pose all possible arguments that avoid revealing this information, since, if it became public knowledge that current algorithms are vulnerable, stronger algorithms will be developed that they would be unable to break.

Such a heuristic can be useful in arguments between computer agents too. Revealing too much information in a current dialogue might damage an agent's chances of winning a future argument.

In the next section, we examine existing approaches to strategy selection, after which we provide the required theoretical foundations for our approach and informally describe it. Section 3 presents our heuristic in a more formal manner. After presenting an illustrative example, we conclude the paper by looking at possible directions in which this work can be extended.

2 Background and Related Research

Argumentation researchers have recognised the need for argument selection strategies for a long time. However, the field has only recently started receiving more attention. Moore, in his work with the DC dialectical system [8], suggested that an agent's argumentation strategy should take three things into account:

– Maintaining the focus of the dispute.
– Building its point of view or attacking the opponent's one.
– Selecting an argument that fulfils the previous two objectives.

The first two items correspond to the military concept of a strategy, i.e. a high level direction and goals for the argumentation process. The third item corresponds to an agent's tactics. Tactics allow an agent to select a concrete action that fulfils its higher level goals. While Moore's work focused on natural language argument, these requirements formed the basis of most other research into agent argumentation strategies.

In 2002, Amgoud and Maudet [9] proposed a computational system which would capture some of the heuristics for argumentation suggested by Moore.

Their system requires very little from the argumentation framework. A preference ordering is needed over all possible arguments, and a level of prudence is assigned to each agent. An argument is assigned a strength based on how convoluted a chain of arguments is required to defend it from attacks by other arguments. An agent can then have a “build” or “destroy” strategy. When using the build strategy, an agent asserts arguments with a strength below its prudence level. If it cannot build, it switches to a destroy strategy. In this mode, it attacks an opponent’s arguments whenever it can. While the authors note other strategies are reasonable, they make no mention of them. Shortcomings of their approach include its basis on classical propositional logic and the assumption of unbounded rationality; computational limits may affect the arguments agents decide to put forth. Finally, no attempt is made to capture the intuition that a fact defended by multiple arguments is more acceptable than one defended by fewer (the so called “accrual of evidence” argument scheme [10]).

Using some ideas from Amgoud’s work, Kakas et al. [11] proposed a three layer system for agent strategies in argumentation. The first layer contains “default” rules, of the form *utterance* $\leftarrow$ *condition*, while the two higher layers provide preference orderings over the rules. Assuming certain restrictions on the rules, they show that only one utterance will be selected using their system, a trait they refer to as determinism. While their approach is able to represent strategies proposed by a number of other techniques, it does require hand crafting of the rules. No suggestions are made regarding what a “good” set of rules would be.

In [12], Amgoud and Prade examined negotiation dialogues in a possibilistic logic setting. An agent has a set of goals it attempts to pursue, a knowledge base representing its knowledge about the environment, and another knowledge base which is used to keep track of what it believes the other agent’s goals are. The authors then present a framework in which these agents interact which incorporates heuristics for suggesting the form and contents of an utterance, a dialogue game allowing agents to undertake argumentation, and a decision procedure to determine the status of the dialogue. Their heuristics are of particular interest as they are somewhat similar to the work we investigate here. One of their heuristics, referred to as the criterion of partial size, uses as much of an opponent’s knowledge as possible, while the heuristic referred to as the criterion of total size attempts to minimise the length of an argument. Apart from operating in a negotiation rather persuasion setting, their heuristics do not consider the amount of information revealed from one’s own knowledge base.

Cayrol et al. [13] have investigated a heuristic which, in some respects, is similar to ours. In their work, an agent has two types of arguments in its knowledge base. The first, referred to as unrestricted arguments, is used as necessary. The second type, consisting of so called restricted arguments, is only used when necessary to defend unrestricted arguments. They provide an extension of Dung’s argumentation framework which allows one to determine extensions in which a minimal amount of restricted knowledge is exposed, thus providing a reasoning procedure representing minimum information exposure. As we discuss in Section

5, argumentation frameworks based on Dung's work leave arguments as very abstract entities, making it difficult to apply the framework to some situations. Furthermore, unlike the work detailed in this paper, Cayrol et al. do not present a dialogical setting in which the heuristic can operate. Also, since their restricted arguments can only be used to defend unrestricted arguments, it is not clear how their heuristic will function in situations where all knowledge is restricted.

In [14], Bench-Capon describes a dialogue game based on Toulmin's work. He identifies a number of stages in the dialogue in which an agent might be faced with a choice, and provides some heuristics as to what argument should be advanced in each of these cases. Only an informal justification for his heuristics is provided.

Apart from guiding strategy, heuristics have seen other uses in dialogue games. Recent work by Chesñevar et al. [15] has seen heuristics being used to minimise the search space when analysing argument trees. Argument schemes [16] are well used tools in argumentation research, and can be viewed as a form of heuristic that guides the reasoning procedure.

3 The Framework and Heuristic

In many realms of argument, auxiliary considerations (apart from simply winning or losing the argument) come into play. In many scenarios, one such consideration is to minimise the information provided to other parties. For example, in a court case between a government and some alleged terrorists, the government might not be willing to reveal the sources of some of its evidence. We thus propose a simple heuristic to guide an agent in a dialogue: when faced with a number of possible arguments to put forth, the one that should be advanced is the one that exposes as little of the agent's internal knowledge as possible. Many extensions and refinements to this heuristic are possible, some of which are discussed in Section 5. However, in this paper we focus on the most simple form of the heuristic for the sake of perspicaciousness.

In formalising our heuristic, we borrow many ideas from other formal argumentation systems (e.g. [17–20]).

We formalise our system in two parts. First we specify the argumentation system itself, and then the heuristic is described, on the basis of this argumentation system.

3.1 The Argumentation Framework

Argumentation takes place over the language Σ, which contains propositional literals and their negation.

Definition 1. ***Argument*** *An argument is a pair* (P, c)*, where* $P \subseteq \Sigma \cup \{\top\}$ *and* $c \in \Sigma$ *such that if* $x \in P$ *then* $\neg x \notin P$*. We define* $Args(\Sigma)$ *to be the set of all possible arguments in our language.*

P represents the premises of an argument (also referred to as an argument's support), while c stands for an argument's conclusion. Informally, we can read an argument as stating "if the conjunction of its premises holds, the conclusion holds". Facts can be represented using the form $(\top, a)$.

Arguments interact by supporting and attacking each other. Informally, when an argument attacks another, it renders the latter's conclusions invalid.

Definition 2. ***Attack*** *An argument* $A = (P_a, c_a)$ *attacks* $B = (P_b, c_b)$ *if* $c_a = \neg c_b$ *or* $\exists f \in P_b$ *such that* $f \equiv \neg c_a$*. For convenience, we write this as* $attacks(A, B)$*.*

An argument is only relevant to an instance of argumentation if its premises are true. We call such an argument *justified*. However, a simple definition of this concept can cause problems when it comes to self attacking (or self defending) arguments, as well as circular reasoning, and care must thus be taken when describing this concept. Before doing so, we must (informally) describe the proof theory used to determine which literals and arguments are in effect at any time.

The idea behind determining what arguments and literals are admissible at any time is as follows. We start by looking at the facts, and determining what knowledge can be derived from them by following chains of argument. Whenever a conflict occurs (i.e. we are able to derive both x and $\neg x$), we remove these literals from our derived set. Care must be taken to also get rid of any arguments (and further facts) derived from any conflicting literals. To do this, we keep track of the conflicting literals in a separate set, whenever a new conflict arises, we begin the knowledge determination process afresh, never adding any arguments whose conclusions are in the conflicting set to the knowledge set. The philosophical and practical ramifications of this approach will be discussed in Section 5.

More formally, an instance of the framework creates two sets $J \subseteq Args(\Sigma)$ and $C \subseteq \Sigma$ representing justified arguments and conflicts respectively.

Definition 3. ***Derivation*** *An argument* $A = (P_a, c_a)$ *is derivable from a set* S *given a conflict set* C *(written* $S, C \vdash A$*) iff* $c_a \notin C$ *and* $(\forall p \in P_a : (\exists s \in S$ *such that* $s = (P_s, p)$ *and* $p \notin C)$ *or* $P_a = \{\top\})$*.*

Clearly, we need to know what elements are in C. Given a knowledge base of arguments $\kappa \subseteq Args(\Sigma)$, this can be done with the following reasoning procedure:

$$J_0 = \{A | A \in \kappa \text{ such that } \{\}, \{\} \vdash A\}$$
$$C_0 = \{\}$$

Then, for $i > 0, j = 1 \ldots i$, we have:

$$C_i = C_{i-1} \cup \{c_A, \neg c_A | \exists A = (P_A, c_A), B = (P_B, \neg c_A) \in J_{i-1} \text{ such that } attacks(A, B)\}$$

$$X_{i0} = \{A | A \in \kappa \text{ and } \{\}, C_i \vdash A\}$$

$X_{ij} = \{A|A \in \kappa \text{ and } X_{i(j-1)}, C_i \vdash A\}$

$J_i = X_{ii}$

The set X allows us to recompute all derivable arguments from scratch after every increment of i[1]. Since i represents the length of a chain of arguments, when $i = j$ our set will be consistent to the depth of our reasoning, and we may assign all of these arguments to J. Eventually, $J_i = J_{i-1}$ (and $C_i = C_{i-1}$) which means there are no further arguments to find. We can thus define the conclusions reached by a knowledge base κ as $K = \{c|A = (P, c) \in J_i\}$, for the smallest i such that $J_i = J_{i+1}$. We will use the shorthand $K(\kappa)$ and $C(\kappa)$ to represent those literals which are respectively derivable from, or in conflict with a knowledge base κ.

We illustrate this algorithm with two examples (not all steps are shown):

Example 1. $\kappa = \{(\top, s), (s, t), (t, \neg s)\}$
$J_0 = \{(\top, s)\}$, $C_1 = \{\}$, $J_1 = X_{11} = \{(\top, s), (s, t)\}$
...
$J_2 = (\top, s), (s, t), (t, \neg s)$
$C_3 = \{s, \neg s\}$
$X_{30} = \{\} \ldots J_4 = J_3 = \{\}$

Example 2. $\kappa = \{(\top, a), (\top, b), (a, c), (b, d), (c, \neg d)\}$
$J_0 = \{(\top, a), (\top, b)\}$
$X_{10} = J_0$, $J_1 = X_{11} = \{(\top, a), (\top, b), (a, c), (b, d)\}$
...
$J_2 = X_{22} = \{(\top, a), (\top, b), (a, c), (b, d), (c, \neg d)\}$
...
$C_3 = \{(d, \neg d)\}$,
$J_4 = J_3 = X_{33} = X_{32} = \{(\top, a), (\top, b), (a, c)\}$

3.2 The Dialogue Game and Heuristic

Agents engage in a dialogue using the argumentation framework described above in an attempt to persuade each other of certain facts. An agent has a private knowledge base (KB) as well as a goal literal g. The environment, apart from containing agents, contains a public knowledge base which takes on a role similar to a global commitment store[2], and is thus referred to as CS.

Definition 4. ***Environment and agents*** *An Agent $\alpha \in Agents$ is a triple $(Name, KB, g)$ where $KB \subseteq Args(\Sigma)$ and $g \in \Sigma$. Name is a unique label assigned to the agent. Given n agents in the system, we assume they are labelled $Agent_0 \ldots Agent_{n-1}$.*

The environment is a pair $(Agents, CS)$ where Agents is the set of agents participating in the dialogue and $CS \subseteq Args(\Sigma)$

[1] This allows us to get rid of long invalid chains of arguments, as well as detect and eliminate arbitrary loops.

Agents take turns to put forward a line of argument consisting of a number of individual arguments. For example, an agent could make the utterance $\{(\top, a), (a, b)\}$. Alternatively, an agent may pass. The dialogue ends when CS has remained unchanged for n turns i.e. after all players have had a chance to modify it, but didn't (this is normally caused by all agents having passed consecutively). Once this has happened, the acceptable set of arguments is computed over the CS, and the status of each agent's goal can be determined, allowing one to compute the winners of the game.

Definition 5. ***Turns and utterances*** *The function*

$$turn : Environment \times Name \to Environment$$

takes an environment and an agent label, and returns a new environment containing the result of the utterance ($utterance : Environment \times Name \to 2^{Args(\Sigma)}$*) made by the labelled agent during its turn.*

$$turn(Environment, \alpha) = (Agents, \{CS \cup utterance(Environment, \alpha)\})$$

During turn i, we will set $\alpha = Agent_{i \bmod n}$, where n is the number of agents taking part in the dialogue. We will detail the *utterance* function for a rational agent below. Before doing so, we define the dialogue game itself. Each turn in the dialogue game results in a new public commitment store, which can be used by the agents in later turns.

Definition 6. ***Dialogue game*** *The dialogue game is defined as*
$turn_0 = turn((Agents, CS_0), Agent_0)$
$turn_i = turn(turn_{i-1}, Agent_{i \ mod \ n})$ *for* $i = 1, 2, \ldots$

The game ends when $turn_i \ldots turn_{i-n+1} = turn_{i-n}$.

CS_0 is dependent on the system, and contains any arguments that are deemed to be common knowledge. Also, note that the null utterance $\{\}$ is defined to be a pass.

By using the derivation procedure described in the previous section, agents can

- Determine, by looking at CS, what literals are in force and in conflict.
- Determine, by combining CS with parts of their own knowledge base, what literals they can prove (or cause to conflict).

By doing the latter, together with looking at the number of literals introduced into K and C, an agent can both determine how much information it reveals by putting forth an argument, and narrowing down the range of possible arguments it will submit (though possibly not to a unique argument).

An agent's first goal is to win the argument by proving its point. If it cannot do so, it will try to obtain a draw. Winning an argument requires that $g \in K(CS)$, while a draw results if no conclusions can be reached regarding the status of g, i.e. $g \in C(CS)$ or $\{g, \neg g\} \cap K(CS) = \{\}$.

Definition 7. ***Winning arguments*** *An agent $\alpha = (Name, KB, g)$ has a set of winning arguments defined as*
$Win = \{A \in 2^{KB} | \ g \in K(A \cup CS)$ *and if* $A \neq \{\}, \{\} \notin A\}$

Definition 8. ***Drawing arguments*** *An agent $\alpha = (Name, KB, g)$ has a set of drawing arguments defined as*
$Draw = \{A \in 2^{KB} | \ (g \in C(A \cup CS)$ *or* $\{g, \neg g\} \cap K(A \cup CS) = \{\})$ *and if* $A \neq \{\}, \{\} \notin A\}$

An *information aware* agent is one that attempts to win an argument while minimising the amount of information it exposes.

Definition 9. ***Information exposure*** *The information exposed by an agent $\alpha = (Name, KB, g)$ making an utterance $A \in 2^{KB}$ can be defined as follows:*

$$Inf = |K(A \cup CS) + C(A \cup CS)| - |K(CS) + C(CS)|$$

Where $K(X)$ and $C(X)$ are the sets of literals obtained by running the reasoning process over the set of arguments X.

An agent prefers a winning strategy over one which leads to a draw, and orders its winning strategies by the amount of information they reveal. This may still lead to multiple possible arguments, in which case other heuristics (such as choosing the shortest possible chain of arguments) may be employed to select a unique argument. We do not discuss these other heuristics in this paper. This preference over arguments can be captured in the following definition:

Definition 10. ***Possible arguments*** *The set of possible arguments an agent would utter is defined as*

$$PA = \begin{cases} A \in Win \ s.t. \ Inf(A) = min(Inf(B)), B \in Win. & Win \neq \{\} \\ A \in Draw \ s.t. \ Inf(A) = min(Inf(B)), B \in Draw & Win = \{\}, \\ & Draw \neq \{\} \\ \{\} & Win = \{\}, \\ & Draw = \{\} \end{cases}$$

The utterance an agent makes is one of these possible arguments: utterance $\in PA$

It should be noted that a "pass", i.e. $\{\}$ might still be uttered as part of the Win or Draw strategy.

When the game is over, all that remains to be done is determine who (if anyone) won the argument:

Definition 11. ***Victory conditions*** *The set of winning agents is $Agents_{win} = \{\alpha = (Name, KB, g) \in Agents | \ g \in K(CS)\}$. Similarly, the set of drawing agents is $Agents_{draw} = \{\alpha = (Name, KB, g) \in Agents | \ g \in C(CS)$ or $\neg g \notin K(CS)$ and $\alpha \notin Agents_{win}\}$. All other agents are in the losing set: $Agents_{lose} = \{\alpha \in Agents | \alpha \notin (Agents_{win} \cup Agents_{draw})\}$*

Literals in $K(CS)$ at the end of the game are those agreed to be in force by all the agents.

In this section, we have defined an argument framework which allows an agent to determine which arguments are in force by performing forward chaining on a knowledge base of arguments, beginning with those arguments which have no premises. We then described a simple dialogue together with a reasoning procedure which allows an agent to put forth arguments revealing as little information as possible. During each move, an agent picks which arguments to reveal from its private knowledge base by computing what literals are in conflict (via $C(CS)$) and which literals would be deemed accepted (by using $K(CS)$) for the new CS containing the arguments it would put forth. If it determines that there are a number of possible arguments it could submit that would win (or, if no winning arguments exist, draw) it the game, it chooses to utter the set of arguments which minimise the amount of information it reveals[2].

Having defined our system, we can now look at its features. In the next section we provide a small example of a dialogue, after which we provide a more in-depth discussion of the framework, heuristic, and features that emerge by studying the example.

4 Example

To increase readability, we present our example in a somewhat informal manner. The argument focuses on the case for, or against, the possibility of weapons of mass destruction (WMDs) existing at some location.

We assume a two party dialogue (with $Agent_0 = \alpha$, $Agent_1 = \beta$), and describe only one agent's knowledge base. At the start of the game, our agent has the following facts in its private knowledge base KB:

$(\top, Chemicals)$	Chemicals exist
$(\top, Photo)$	Photos exist
$(\top, Newspaper)$	Newspaper articles exist
$(\top, Factory)$	Factories exist
$(\top, \neg Medicine)$	Medicine is not being produced
$(Newspaper, WMD)$	If newspapers say so, then WMDs exist
$(\{Photo, Factory\}, \neg WMD)$	Pictures of factories mean WMDs don't exist
$(Chemicals, \neg Medicine)$	Chemicals mean medicine isn't being produced
$(\{Chemicals, Factory\}, WMD)$	Chemicals and factories mean WMDs exist

[2] A Prolog implementation of the argumentation framework, dialogue game and heuristic is available at `http://www.csd.abdn.ac.uk/~noren`

Then the following dialogue takes place (α's goal is the literal WMD):

$$\begin{array}{lr}
(\alpha)\ (\top, Newspaper), (Newspaper, WMD) & 1 \\
(\beta)\ (\top, \neg Newspaper), (\top, Factory), (Factory, Medicine), & \\
\quad (Medicine, \neg WMD) & 2 \\
(\alpha)\ (\top, Chemicals), (Chemicals, \neg Medicine), & \\
\quad (\{Chemicals, Factory\}, WMD) & 3 \\
(\beta)\ \{\} & 4 \\
(\alpha)\ \{\} & 5
\end{array}$$

Informally, agent α claims that since newspaper articles about the subject exist, WMDs must exist (as per the newspaper's claims). β responds by saying that it has not seen any articles, but that since he knows that factories exist, and that these factories produce medicines, WMDs are not present (possibly implying that any evidence found is due to these medicines). α counters that chemicals were found, and that the finding of these is incongruent with the presence of medicines, also stating that the presence of the factories and the chemicals is proof regarding the existence of WMDs. β has no response to this, and after α stays silent, the game ends with α successfully proving his goal.

Before examining the dialogue in detail, we can discuss a few interesting, global properties of the heuristic:

- An agent that knows it will lose an argument is still able to win it by assuming that its opponent does not have access to the same information it does. It could be argued that passing to draw (or win) a game when it has not revealed all its information is tantamount to lying.
- The heuristic is different to the "Occam's razor" heuristic that has often appeared in the literature. The latter proposes that the shortest argument be put forth first, while we are able to present longer arguments if they reveal less information. In many cases however, the two heuristics can coincide regarding what utterance should be made next.

Let us examine line 3 in more detail. Before this line, our public knowledge base, CS, contained the following arguments:

$$\begin{array}{lll}
(\top, Newspaper) & (Newspaper, WMD) & (\top, \neg Newspaper) \\
(\top, Factory) & (Factory, Medicine) & (Medicine, \neg WMD)
\end{array}$$

Clearly, apart from having an information exposure value (*Inf* in Definition 9) of one, an argument such as $(\top, Photo)$ will not be considered as it is not part of the winning or drawing set. The argument chosen has an information exposure value of two (as the literals $Chemicals$ and $\neg Medicine$ are added to CS), but was chosen as it is part of the winning set. Note that an argument such as

$$\begin{array}{ll}
(Chemicals, \neg Medicine), & (\top, Chemicals) \\
(\{Chemicals, Factory\}, WMD), & (\top, Photo)
\end{array}$$

is also part of the winning set, but has a higher information exposure value.

The argument $(Chemicals, \neg Medicine), (\top, Chemicals)$ Belongs to the drawing set, and has an information exposure value of one.

The argument $(\top, Photo), (\{Photo, Factory\}, \neg WMD)$ has an information exposure value of 2, and, if suggested after line 3 of the dialogue, is part of the drawing set. It will thus not be selected as an utterance.

Since our winning set is non-empty, our agent was forced to pick an argument from there. By modifying Definition 10, we could define a number of different classes of agents with a range of preferences based on winning, drawing or losing an argument and revealing different amounts of information.

5 Discussion

This section examines the argumentation framework and the heuristic, tying it back to the concept of an argumentation strategy as proposed by Moore. We also examine some of the novel features of argument that emerge when dialogue takes place in the framework using the heuristic, and propose avenues for future research.

Our approach seems to share much in common with the "sceptical" approach to argumentation. When arguments conflict, we refuse to decide between them, instead ruling them both invalid. This means that our reasoning procedure is not complete, given the (rather convoluted) set of arguments

$$(\top, A), (\top, B), (A, \neg B), (B, \neg A), (A, C), (B, C), (\neg A, C), (\neg B, C)$$

we see that C should hold, but doesn't. Other argumentation systems (namely those utilising the unique–status–assignment approach [4]) are similarly incomplete, leaving this an open area for future research. Our sceptical approach does yield a consistent system, as no conflicting arguments will remain in the final set of arguments.

The simplicity of our approach means that only specific types of arguments can be represented (namely, those whose premises are a conjunction of literals, and whose conclusion is a single literal). However, as seen in the example, even with this limitation, useful arguments can still emerge.

We developed our own argumentation framework rather than using an existing one for a number of reasons, including:

- The abstract nature of many frameworks (e.g. [17]) makes arguments atomic concepts. We needed a finer level of granularity so as to be able to talk about which facts are exposed (allowing us to measure the amount of information revealed during the dialogue process). Less abstract frameworks (e.g. [21, 18]), while looking at concepts such as derivability of arguments still primarily focus on the interactions between arguments.
- Almost all other frameworks define higher level concepts in terms of arguments attacking, defeating and defending one another. For us, the concept of one argument justifying another is critical, together with the concept of attack.

– Other argumentation systems contain concepts which we do not require, such as a preference ordering over arguments.
– Approaches such as [18] divide their argument constructs into defeasible and indefeasible sets, with a consistency requirement on the indefeasible set, and then provide for default reasoning over the defeasible arguments. Our framework only takes the defeasible nature of arguments into account, ignoring default reasoning.

While representing the heuristic using one of the other approaches is (probably) not impossible, it appears to be more difficult than by using our own system.

Looking at Moore's three criteria for an agent argumentation strategy, we see that our heuristic fulfils its requirements. If the focus of the argument were not maintained, more information would be given than is strictly necessary to win, thus fulfilling the first requirement. Both the second and third requirements are clearly met by the decision procedure for which argument to advance described in Definition 10.

Investigating the use of the heuristic in more complex settings (by either increasing the representational power of the framework, or by representing the heuristic in another argumentation framework) is one possible direction of future work.

One disadvantage of our approach is that at each move, we evaluate possible arguments from the powerset of an agent's private knowledge. This leads to an exponential complexity in our algorithm. While simple techniques can be applied to shrink the size of the powerset, more complicated approaches which can further reduce the algorithm's running costs need to be examined.

Making the heuristic more realistic is another area we are investigating. For example, rather than treating all information equally, we could assign a numerical cost to each literal, and attempt to minimise this cost while winning the argument. Another avenue for future research involves determining how this heuristic can best be combined with techniques for resource bounded reasoning. Allowing agents to communicate with each other privately, rather than with all dialogue participants allows for a number of knowledge bases to exist. An agent might have certain information it is willing to reveal to some, but not all participants, and investigating strategies for such dialogues is another rich research area.

6 Conclusions

In this paper we proposed a heuristic for argumentation based on revealing as little information as possible to the other dialogue participants. While such an argumentation strategy arises in many real world situations, we are not familiar with any application that explicitly makes use of this technique. To study the heuristic, we proposed an argumentation framework that allowed us to focus on it in detail. Several novel features emerged from the interplay between the heuristic and the framework, including the the ability of an agent to win an argument that it should (given all possible information) not be able to win. While we have only examined a very abstract model utilising the heuristic, we believe that many interesting extensions are possible, and many unanswered questions remain.

Acknowledgements

This work is partly funded by the DTI/EPSRC E-Science Core Program and British Telecom, via a grant for the CONOISE-G project (http://www.conoise.org), a multi-university collaboration between Aberdeen, Cardiff and Southampton universities, and BT. We would also like to thank Madalina Croitoru for her useful feedback on earlier versions of this paper.

References

1. Prakken, H., Sartor, G. In: Computational Logic: Logic Programming and Beyond. Essays In Honour of Robert A. Kowalski, Part II. Volume 2048 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2002) 342–380
2. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue. State University of New York Press (1995)
3. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Computing Surveys **32**(4) (2000) 337–383
4. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In Gabbay, D., Guenthner, F., eds.: Handbook of philosophical logic, 2nd Edition. Volume 4. Kluwer Academic Publishers (2002) 218–319
5. Walton, D.N.: Legal argumentation and evidence. Penn State Press (2002)
6. McBurney, P., Parsons, S.: Risk agoras: Dialectical argumentation for scientific reasoning. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, Stanford, USA (2000) 371–379
7. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., New York, NY, USA (1993)
8. Moore, D.: Dialogue game theory for intelligent tutoring systems. PhD thesis, Leeds Metropolitan University (1993)
9. Amgoud, L., Maudet, N.: Strategical considerations for argumentative agents (preliminary report). In: NMR. (2002) 399–407
10. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proceedings of the tenth International Conference on Artificial Intelligence and Law. (2005) 85–94
11. Kakas, A.C., Maudet, N., Moraitis, P.: Layered strategies and protocols for argumentation-based agent interaction. In: ArgMAS. (2004) 64–77
12. Amgoud, L., Prade, H.: Reaching agreement through argumentation: a possiblistic approach. In: Proceedings of KR 2004. (2004)
13. Cayrol, C., Doutre, S., Lagasquie-Schiex, M.C., Mengin, J.: " minimal defence" : a refinement of the preferred semantics for argumentation frameworks. In: Proceedings of NMR-2002. (2002)
14. Bench-Capon, T.J.: Specification and implementation of Toulmin dialogue game. In: Proceedings of JURIX 98. (1998) 5–20

15. Chesñevar, C.I., Simari, G.R., Godo, L.: Computing dialectical trees efficiently in possibilistic defeasible logic programming. In Baral, C., Greco, G., Leone, N., Terracina, G., eds.: Logic Programming and Nonmonotonic Reasoning, 8th International Conference, LPNMR 2005, Diamante, Italy, September 5-8, 2005, Proceedings (LNCS 3662). Lecture Notes in Computer Science, Springer (2005) 158–171
16. Reed, C.A., Walton, D.N.: Applications of argumentation schemes. In Hansen, H.V., Tindale, C.W., Blair, J.A., Johnson, R.H., Pinto, R.C., eds.: Proceedings of the 4th Conference of the Ontario Society for the Study of Argument (OSSA2001), Windsor, Canada. (2001) CD ROM
17. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2) (1995) 321–357
18. Prakken, H., Sartor, G.: A dialectical model of assessing conflicting arguments in legal reasoning. Artificial Intelligence and Law **4** (1996) 331–368
19. Pollock, J.L.: Perceiving and reasoning about a changing world. Computational Intelligence **14** (1998) 498–562
20. Verheij, B.: DefLog: On the logical interpretation of prima facie justified assumptions. Journal of Logic and Computation **13**(3) (2003) 319–416
21. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artif. Intell. **53**(2-3) (1992) 125–157