Opponent Modelling in Persuasion Dialogues

Christos Hadjinikolis, Yiannis Siantos, Sanjay Modgil, Elizabeth Black, Peter McBurney

King’s College London, Department of Informatics
London, UK
name.surname@kcl.ac.uk

Abstract

A strategy is used by a participant in a persuasion dialogue to select from the possible locutions it may make, one which is most likely to achieve its objective of persuading its opponent. Such strategies often assume that the participant has a model of its opponents, which may be constructed on the basis of a participant’s accumulated dialogue experience. However, in most cases the fact that an agent’s experience may encode additional information which if appropriately used could increase a strategy’s efficiency, is neglected. In this work, we rely on an agent’s experience to define a mechanism for augmenting an opponent model with information likely to be dialectically related to information already contained in it. Precise computation of this likelihood is exponential in the volume of related information. We thus describe and evaluate an approximate approach for computing these likelihoods based on Monte-Carlo simulation.

1 Introduction

Agents engaging in persuasion dialogues aim to convince their counterparts to accept a proposition that the last does not currently endorse. In this sense strategising is essential, and as one would expect, it is mainly based on an agent’s assumptions about the beliefs of its opponents. This is widely studied in terms of opponent modelling. Essentially, an opponent model (OM) consists of five basic components: an opponent’s knowledge; abilities; preferences; objectives, and; strategy. Numerous researchers who deal with the knowledge; abilities; preferences; objectives, and; strategy.

The above dialogue is essentially composed of two lines of dispute \{A←B←C\} and \{A←D←E\}. Assume then, that \(A_{Q2}\) engages in a persuasion dialogue with another agent, \(A_{Q3}\), on which is the best restaurant in town. Let us also assume that at some point in the dialogue \(A_{Q3}\) cites the newspaper article, by asserting argument A in the game, as \(A_{Q1}\) did in the previous dialogue. It is then reasonable for \(A_{Q2}\) to expect that to some extent \(A_{Q3}\) is likely to also be aware of the chef’s qualifications (argument C). Intuitively, this implies that consecutive arguments in the same dispute lines of a dialogue share some relationship. In this case, the “chef’s proposition” (A) and “his qualifications” (C) appear to be related in the sense that awareness of the first implies a likely awareness of the second. This is also the case for arguments A and E. However, assuming such a relationship between the chef’s arguments, due to the simplicity of such modelling approaches, in most cases the formalisation of an OM is left implicit. In addition, such approaches disregard the fact that an agent’s accumulated dialogue experience may encode additional information which could also be used for modelling, and so increasing the effectiveness of strategies that rely on OMs.

In this work, we rely on a simple framework for persuasion dialogue with abstract arguments. Participants are assumed to maintain OMs which are constantly updated with content (arguments) obtained through new dialogues. We focus on proposing a mechanism which is used for augmenting such an OM, by adding to it additional information that is likely to be associated with information already contained in it. In other words, we attempt to predict what else is likely to be known to a particular agent, given: a) what we currently assume the latter knows, and; b) what others with similar knowledge know. For doing this we rely on an agent’s general history of dialogues, in which we monitor the times that certain opponent arguments (OAs) follow after certain others, thus utilising an agent’s experience in a multifaceted way. For example, let two agents, \(A_{Q1}\) and \(A_{Q2}\), engage in a persuasion dialogue in order to decide where is the best place to have dinner:

- \(A_{Q1}\): (A) We should go to the Massala Indian restaurant since a chef in today’s newspaper recommended it.
- \(A_{Q2}\): (B) A single chef’s opinion is not trustworthy.
- \(A_{Q1}\): (C) This one’s is, as I have heard that he won the national best chef award this year.
- \(A_{Q2}\): (D) Indian food is too oily and thus not healthy.
- \(A_{Q1}\): (E) It’s healthy, as it’s made of natural foods and fats.

We thus describe and evaluate an approximate approach for computing these likelihoods based on Monte-Carlo simulation.
qualifications (C) and the argument on why Indian food is considered healthy (E), seems counter intuitive, as these arguments belong in different dispute lines, each of them supporting A in a different way. We thus assume that two arguments can be related if they are found in the same dispute lines of a dialogue, where this relationship can be understood in terms of the notion of support, e.g., C supports A against B.

In this sense, relying on an agent’s accumulated dialogue experience, one can define a graph in which links between OAs asserted in a series of dialogues, indicate support. A participant may then rely on this graph in order to augment a current model of a possible opponent, by adding to it arguments which, according to the graph, are linked with arguments already contained in the model, and which are thus likely to be known to that opponent. For quantifying this likelihood we rely on how often a certain argument follows after another in an agent’s history of dialogues. While this is a simple approach, we believe that it is sufficient for increasing the effectiveness of an agent’s strategising. In the future we intend to investigate more complex ways for quantifying this likelihood, accounting for contextual factors such as how common certain information is, or whether an agent is a member of a particular group having access to shared information, etc.

Summarising, while an agent’s dialogue experience can be utilised in many ways when modelling its opponents, in most cases this is neglected. We thus present an applicable method for updating and augmenting such a model utilising an agent’s history of persuasion dialogues, making the following contributions: a) in Section 3, we rely on an argumentative persuasion dialogue framework (formalised in Section 2) to define and present two mechanisms responsible for updating and augmenting an OM; b) in Section 3.1 we define our augmentation mechanism, formalising a method for building a graph relating supporting arguments, based on an agent’s experience; c) in Section 3.1 we also provide a method for augmenting an agent’s current beliefs about its opponents’ beliefs based on these support relations, enabling an agent to additionally account for this information in its strategising; d) in Section 4 we finally define and analyse a Monte-Carlo simulation approach concerned with the augmentation process, which makes our approach tractable. We also prove converge and provide supporting experimental results.

2 The Dialogue Framework

In this section we describe an argumentative system for persuasion dialogue. Essentially, such a system relies on an argumentation framework (AF). In general an AF is a pair \( AF = (A, C) \) where \( A \) is a set of arguments and \( C \) is a binary relation on \( A \), i.e. \( C \subseteq A \times A \), while for two arguments \( A \) and \( B \), \( (A, B) \in C \) means that \( A \) represents an attack on \( B \) [2].

We assume that utterances exchanged in dialogues, such as “\( x \) since \( y \)”, are abstractly represented as arguments, while if a participant questions or contradicts this argument, by either asking “\( \text{why } y \)" or saying “\( \text{not } y \)" respectively, then this can be perceived as an attack. We assume a general framework for persuasion dialogue where the participants, a proponent and an opponent, debate the truth of an argument \( X \) (topic), through exchanging dialogue moves (DMs), consisting of arguments, based on the attack relationship between them, and on a set of protocol rules that regulate the dialogue process; i.e. a participant can introduce an argument into the game if it attacks another argument that was previously introduced into the game by its interlocutor. We assume that the participants share the same language \( L \), and that there is agreement as to whether a given argument attacks another.

In this respect, we define a dialogue \( D \) as a sequence of dialogue moves \( <DM_0, \ldots, DM_n>_D \), where the content of \( DM_0 \) is an argument for \( X \), which defines the topic of the dialogue. We further assume that the introduction of moves is contingent upon satisfying certain conditions, defined by a dialogue protocol, which concern: turntaking; backtracking; the legality of a move, and; the game’s termination rules.

Turntaking specifies the participant to move next and the number of dialogue moves she can make. We assume agents alternate turns introducing a single move at a time. Backtracking (otherwise referred to as multi-reply), which we also assume for our protocol, concerns whether a participant is allowed to return to a previous point in the game and respond to a previous move of its interlocutor in a different way. By licensing backtracking, it is easy to see that a dialogue \( D \) can be represented as a tree \( T \), which we define as follows:

**Definition 1.** Let \( D = <DM_0, \ldots, DM_n>_D \) be a dialogue and \( M = \{DM_0, \ldots, DM_n\} \) the set of all moves employed in \( D \), then \( T = (M, E) \) is a dialogue tree where \( E \subseteq M \times M \) is a set of arcs that associate elements of \( M \), such that:

- for two moves \( DM_1 \) and \( DM_2 \), \( (DM_1, DM_2) \in E \) means that \( DM_2 \) is \( DM_1 \)’s target
- \( DM_0 \) represents \( T \)’s root node
- every move in \( M \) that does not serve as any other move’s target, is a leaf node
- for a number of leaf nodes equal to \( n \), every \( d_i \), for \( 1 \leq i \leq n \), is a distinct path from \( DM_0 \) to one of \( T \)’s leaves, which we denote as dispute.

The set of all disputes found in a tree is represented as \( \Delta = \{d_1, \ldots, d_m\} \). Each new tree path results from the introduction of a backtracking move by either of the participants. Such an example can be found in the dialogue tree illustrated in Figure 1(a), where grey’s move \( DM_0 \) is used as an alternative reply against white’s move \( DM_1 \).

![Figure 1](image-url)
The legality of a dialogue move is concerned with explicit rules related with: the dialogical objective; a participant’s role in it; and its commitments, which we assume to be stored in a commitment store (CS) being constantly updated with the content of the dialogue moves that a participant introduces during the course of the dialogue.

**Definition 2** For an agent \(Ag_i \in Ags\) participating in a dialogue \(\mathcal{D} = \langle DM_{t_0}, \ldots, DM_{t_n} \rangle\), we define its commitment store as a set \(CS_i^t = \{A_i^0, \ldots, A_i^n\}\), for \(k < n\), which contains the arguments introduced into the game by \(Ag_i\) up to turn \(t\), for \(t = 0 \ldots n\), such that \(CS_i^0 = 0\), and:

\[
CS_i^{t+1} = CS_i^t \cup \text{Content}(DM_{t+1})
\]

where \(\text{Content()}\) returns the argument comprised in a \(DM\). We say in this case that information is directly updated.

Finally, we assume that each agent \(Ag_i \in Ags\) can engage in dialogues in which its strategic selection of moves is based on what \(Ag_i\) believes its opponent (\(Ag_{j \neq i}\)) knows. Accordingly each \(Ag_i\) maintains a model of its possible opponents. In a similar sense to [9], the proposed model consists of abstract arguments while we disregard modelling an agent’s goals as the latter is out of the scope of this work.

**Definition 3** Let \(Ags\) be a set of agents \(\{Ag_1, \ldots, Ag_v\}\), then for \(i = 1 \ldots v\), the knowledge base \(KB\) of \(Ag_i\) is a tuple \(KB_i = \langle A_i^0, \ldots, A_i^n\rangle\) such that for \(j = 1 \ldots v\), each sub-base \(A_{i,j} = \{A_1, \ldots, A_k\}\), where \(k \in \mathbb{N}\), is a set representing an OM expressing what \(Ag_i\) believes is \(Ag_j\)'s known arguments; and where \(A_{i,j}\) represents \(Ag_i\)'s own knowledge.

### 3 Modelling mechanisms

For modelling an agent’s opponents’ beliefs we rely on the arguments they put forth in a dialogue game, assuming at the same time that agents believe what they utter. We assume so since we acknowledge that it is not possible to impose that agents are truthful through protocol restrictions or regulations [13]. We believe that the proposed approach will be very effective if employed in accordance with trust related semantics [14], something we intend to investigate in the future.

We begin by associating a confidence value \(c\) to the arguments of a sub-base \(A_{i,j}\). For an agent \(Ag_i\), this value expresses the probability of a certain argument in \(A_{i,j}\) being part of \(Ag_j\)'s actual knowledge \(A_{j,j}\). For the computation of this value we differentiate between whether information is: a) gathered directly by \(Ag_i\), on the basis of its opponent’s updated CS, or; b) a result of an augmentation attempt of \(Ag_i\)'s current model of \(Ag_j\). The latter concerns an incrementation of a current OM with the addition of arguments that are likely to also be known to \(Ag_j\)’s opponent.

As noted in Section 1, intuitively we expect our opponents to be aware of arguments that are likely to follow in a current dialogue, given that they have appeared in previous dialogues, and relate to what we currently assume our opponent to know. We assume this likelihood to increase as the relation between the contents of an OM and the arguments external to the model becomes stronger. This is due to the appearance of particular sequences of arguments in dialogues, which relate the two sets. For example, assume \(Ag_i\) (\(P\)) and \(Ag_j\) (\(O\)) engage in a persuasion dialogue, where backtracking is allowed. Let us further assume that the dialogue tree illustrated in Figure 1(a) describes such a dialogue. In this case \(Ag_i\) and \(Ag_j\) introduce arguments \(\{A, C, E, G\}\) respectively \(\{B, D, F, H\}\). Assume then, that \(Ag_i\) engages in another persuasion dialogue with a different agent \(Ag_m\) who also happens to counter \(Ag_i\)'s \(A\) with \(B\). It is then reasonable to assume that \(Ag_m\) is likely to be aware of arguments \(D, H\) or even \(F\). In this respect, one may assume an implied support relationship between between these arguments. If then \(Ag_m\) does indeed put forth arguments \(D, H\) and \(F\) in the game, then the likelihood of someone knowing \(D, H\) and \(F\), contingent that she knows \(B\), should increase.

For assigning a confidence value \(c\) to the elements of an \(A_{i,j}\), we assume that every agent retains its own arguments without revision. In other words, we assume that introduction of conflicting arguments does not cause any older arguments to be discarded, but rather conflicts are decided by some non-monotonic inference mechanism, i.e., an agent’s beliefs may be warranted by justifying arguments under argumentation acceptability semantics [2]. We therefore assume that the confidence value of information acquired directly from the commitment store of one’s opponent is equal to 1, which represents the highest level of confidence.

**Definition 4** Assume an \(A_{i,j} \in KB_i\), then \(\forall X \in A_{i,j}\), there is a tuple \(<X, c>\) where \(c\) is the confidence level of \(X\) such that:

\[
c^{[0,1]} = \begin{cases} 
1 & \text{if } X \text{ is directly collected by } Ag_i \\
Pr(X) & \text{if } X \text{ is part of an augmentation of } A_{i,j}
\end{cases}
\]

where \(Pr(X)\) is the likelihood of \(X\) being also known to \(Ag_j\), determined partly by the current form of \(A_{i,j}\).

Further details on determining \(Pr(X)\) follow in Section 3.1. For providing the mechanism responsible for updating an agent’s \(A_{i,j}\) we need to first define the notion of history.

**Definition 5** Let \(Ags\) be a set of agents \(\{Ag_1, \ldots, Ag_v\}\), then for two agents \(Ag_i\) and \(Ag_j\), \(i \neq j\), we assume \(h_{i,j} = \{\mathcal{G}_1, \ldots, \mathcal{G}_\nu\}\) to be \(Ag_i\)'s history of dialogues with \(Ag_j\). Accordingly, we assume \(H_i = \bigcup_{j \neq i} h_{i,j}\) be the set of all histories of \(Ag_i\) with each member \(Ag_j \in Ag_s\).

Opponent information is then updated as follows:

**Definition 6** Let \(Ag_i\) and \(Ag_j\) be two agents in \(Ags\), then, given the current version of a sub-base \(A_{i,j}^{\mu-1}\) and \(Ag_j\)'s commitment store \(CS_j\) of the latest dialogue \(\mathcal{G}_\nu\), \(Ag_i\) can update its sub-base as follows:

\[
A_{i,j}^{\mu} = A_{i,j}^{\mu-1} \cup CS_j
\]

As explained, in this case information is directly collected and thus is given a confidence value equal to 1.

### 3.1 Building a \(RG\) & augmenting the OM

For augmenting a \(A_{i,j}\), we rely on a relationship graph (\(RG\)). For an \(Ag_i\) a relationship graph (\(RG_i\)) is a graph that associates nodes, that represent arguments asserted by an agent’s opponents in \(H_i\). This association is represented
through weighted directed arcs which represent a support relationship between arguments, while the weight represents the likelihood of that relationship, i.e. the extent to which two arguments are related. We assume an \( \mathcal{RG} \) to be incrementally built as an agent \( A_t \) engages in numerous dialogues, being empty at the start, i.e. for \( \mathcal{H}_t = \{ \emptyset \} \), and constantly updated with newly encountered opponent arguments (OAs). In the case of the example presented in Figure 1(a), assuming that the grey agent is the modeller, notice that OAs (the white’s arguments \( B, D, F \) and \( H \)) can only appear in odd levels of the dialogue tree. For assigning arcs between these arguments one may rely on how and when an argument appears in a tree.

Essentially, we assume two OAs, \( X \) and \( Y \), to be connected in a \( \mathcal{RG} \) if they are found in the same path of the dialogue tree, and are of \( d \)-hop distance from each other—distance is measured disregarding the modeller’s DMs. For example, again in Figure 1(a) arguments \( B \) and \( D \) are assumed to be at 1-hop distance from each other, while \( B \) and \( F \) are at a 2-hop distance. Figures 1(b) and 1(c) illustrate two distinct \( \mathcal{RG}s \) induced from the dialogue tree of Figure 1(a), for \( d = 1 \) and \( d = 2 \) respectively. Intuitively, this modelling approach captures the implied relationship that consecutive OAs have in a single branch of a tree. Through modifying the \( d \) (distance) value one can strengthen or weaken the connectivity, between arguments in the same path of a \( T \), and correspondingly between arguments in the induced \( \mathcal{RG} \). Of course, assigning a large number to \( d \) could raise a cognitive resources issue, due to the large volume of information that needs to be stored. An example of incrementally building a \( d = 1 \ \mathcal{RG} \) after two consecutive persuasion dialogues is illustrated in Figure 2.

In this respect, let \( A_t^{\mathcal{H}_t} \) represent all the arguments introduced by \( A_t \)’s opponents in \( \mathcal{H}_t \), we define a \( \mathcal{RG} \) as follows:

**Definition 7** Assume an agent \( A_t \), then a \( \mathcal{RG} \) is a weighted directed graph \( \mathcal{RG}_t = \{ A_t^{\mathcal{H}_t}, R \} \), whose nodes are represented by elements of set \( A_t^{\mathcal{H}_t} \), and where \( R \subseteq A_t^{\mathcal{H}_t} \times A_t^{\mathcal{H}_t} \) is a set of weighted arcs, each of them indicating a support relationship between two arguments in \( A_t^{\mathcal{H}_t} \). We denote with \( r_{AB} \in R \) an arc that extends from \( A \) to \( B \) where \( A, B \in A_t^{\mathcal{H}_t} \), while we denote the arc’s weight as \( w_{AB} \) retrieved via a weight function \( w \), such that \( w : R \rightarrow [0,1] \).

In order to determine whether two arguments in a \( \mathcal{RG} \) are connected via an arc we rely on the following condition:

**Condition 1** Let \( \Delta = \{ d_1, \ldots, d_k, \ldots, d_n \} \) be the disputes of a dialogue tree \( T \), for every distinct pair of dialogue moves \( DM_1 \) and \( DM_2 \) respectively comprising arguments \( A \) and \( B \), that respectively appear in levels \( i \) and \( j \), for \( i \) and \( j \) being opponent levels (either odd or even depending on who initiated the dialogue) where \( j \geq i \), if: a) \( DM_1 \in d_k \) and \( DM_{j \in d_k} \), and b) \( \frac{j-i}{2} \leq d \); then \( 3 r_{AB} \in R \).

Lastly, for providing a weight value \( w_{AB} \) which will essentially represent the relationship likelihood of an argument \( A \) with an argument \( B \), we rely on Definition 8, which is essentially a normalisation that allows us to compute a probability value \( Pr(r_{AB}) = w_{AB} \) for arc \( r_{AB} \). We simply count the number of agents that have used in dialogues argument \( A \) followed by \( B \), and we put them against the total number of agents that have simply put forth argument \( A \) in dialogues.

**Definition 8** Given an \( A_t \) and its \( \mathcal{RG}_t = \{ A_t^{\mathcal{H}_t}, R \} \), and two arguments \( A, B \) elements of \( A_t^{\mathcal{H}_t} \) then:

\[
\text{Occurrces}(H, A, B) = M_{AB}
\]

is a function that returns a set \( M_{AB} \subseteq A_t \) representing the set of agents that have put forth argument \( A \) followed by \( B \) in the same disputes and at a distance \( d \) in distinct dialogues in \( \mathcal{H}_t \), satisfying Condition 1 such that \( r_{AB} \in R \), then:

\[
w_{AB} = \frac{|M_{AB}|}{|M_A|}
\]

In the case where \( B \) is omitted then \( M_A \) will represent the set of agents that have simply put forth argument \( A \) in distinct dialogues, while it is evident that \( |M_{AB}| \leq |M_A| \).

We should note that there is a problem with the aforementioned approach. Namely, if we consider the example shown in Figure 2, and based on Definition 8, then all the arcs in the induced \( \mathcal{RG} \) will initially have a weight value of 1. It is apparent that this value does not represent the real likelihood which relates the arguments at either endpoint of the arc. In order to better approach the real likelihood a larger number of dialogues with numerous distinct participants need to be considered. This problem is better known as the cold start problem and is encountered in various other contexts as well [5].

Having built a \( \mathcal{RG} \) an agent \( A_t \) can then attempt to augment its OM (\( A_t^{(i,j)} \)) of \( A_t \) by adding to it the possible arguments (nodes) that are of \( d \)-hop distance in \( \mathcal{RG} \) from those contained in \( A_t^{(i,j)} \). In a trivial case, assume an \( \mathcal{RG}_1 \) induced by \( A_t \) as it is illustrated in Figure 3(a), where for presentation convenience we assume that the weights on the arcs have received their values after numerous dialogue interactions. Let us assume that based on \( A_t^{(i,j)} \)’s OM of \( A_t^{(i,j)} \), \( A_t \) believes that \( A_t^{(i,j)} \) is aware of two arguments \( A_t^{(1,4)} = \{ B, H \} \) (the grey nodes in Figure 3(a)). Hence, \( A_t^{(i,j)} \) computes the likelihood of each of the possible augmentations \( A_t^{(1,4)} \in P \) where \( P = \{ A_t^{(2,i)}, A_t^{(2,i)}, A_t^{(2,i)} \} \), as those appear in Figures 4(b)(c)(d) and (e), and selects the one with the highest likelihood for augmenting \( A_t^{(1,4)} \) with additional contents. Computing each of these likelihoods is done as illustrated in the following example:

**Example 1** Assume we want to calculate the likelihood of augmentation \( A_t^{(1,4)} \rightarrow A_t^{(2,i)} \). In this simple example the likelihood of including belief \( F \) is:

\[
Pr(F) = Pr(r_{HF} \cup r_{BF}) = Pr(r_{HF}) + Pr(r_{BF}) - Pr(r_{BF} \cap r_{HF}) = w_{HF} + w_{BF} - w_{BF} \cdot w_{HF} = 0.82
\]
The probability of inducing $A'_F$ therefore is the probability of including argument $F$ and not including $D$ which is:

$$Pr(A'_F) = Pr(F)(1 - Pr(D)) = Pr(F)(1 - w_{BD}) = 0.328$$

Finally, $Pr(F)$ is also used to denote the confidence value $c$ of argument $F$, as defined in Definition 4.

For providing the general formula for computing the likelihood of a possible augmentation we rely on basic graph theory notation with respect to a node $X$ in a graph $RG$, such as degree $d(X)$, neighbouring nodes $N(X)$ where $|N(X)| = d(X)$, and adjacent arcs $R(X)$. We additionally define $N_A$ for a set of arguments $A$ such that:

$$N_A = \bigcup_{X \in A} N(X)\{Y \in N(X) : Y \notin A\}$$

Essentially, set $N_A$ represents the neighbours of the nodes in $A$, and is formed from the union of the neighbours of every node $X$ in $A$, excluding those that are already in $A$, while $R_A$ is the adjacent arcs of the nodes in $A$ and is equal to the adjacent arcs of every element $X$ in $A$, excluding those that connect with arguments already in $A$.

We note that for $A_{(i,j)}$ it reasonably holds that $A_{(i,j)} \subseteq A^{R_X}$, while for convenience we will hence refer to a $A_{(i,j)}$ as $A$ and to its augmentation as $A'$. Given these, let $P = \{A'_0, A'_1, \ldots \}$ be the set of all possible distinct augmentations of $A$, then the number of all its possible distinct expansions with respect to neighbouring nodes that are of 1-hop distance from $A$, is:

$$|P| = \sum_{k=0}^{N_A} \binom{|N_A|}{k}$$

The general formula for computing the likelihood of a possible augmentation $A'$ with respect to the neighbouring nodes (arguments), i.e. for every $X \in N_A$ of a set $A$ is:

$$Pr(A') = \prod_{X \in A'} Pr(X) \prod_{X \notin A'} (1 - Pr(X))$$

$$Pr(X) = Pr(\bigcup_{r \in R_X} r)$$

where $R_X$ (the in-bound arcs) is used to denote the adjacent arcs of $X$ which are of the form $(X,Y)$, i.e. $Y$ is the arc target. Lastly, since the likelihood of each possible augmentation should define a distribution of likelihoods then:

$$\sum_{A' \in P} Pr(A') = 1$$

4 The Monte-Carlo Simulation

A drawback of the proposed approach is that, calculating the probability of Equation 5 is of exponential complexity. This makes the approach practically intractable. However, drawing inspiration from the work of Li et al. [6] we rely on an approximate approach for computing these likelihoods based on a Monte-Carlo simulation. For the case of the 1-hop augmentation we know that the number of possible augmentations is exponential on the size of $N_A$. It is possible to deduce a high likelihood augmentation in linear time, provided we know $Pr(Y)$ for $\forall Y \in N_A$. Also, $Pr(Y)$ is calculated through Equation 5 using the inclusion-exclusion principle found in basic algorithm textbooks such as [4]. However, this is of exponential complexity on the in-degree to calculate. We therefore proceed to sample for $Pr(Y)$, by describing a method to sample for high likelihood arguments which we will include in our 1-hop augmentation of a set $A$. We will generally refer to the arguments in $A$ as augmentation nodes. The method we describe is as follows:

- Assume an $RG = \{A^N, R\}$ and a set $A$
- We begin with a set of nodes $A' = A$
- For each argument $Y \in N_A$ and if $\exists r_{XY} \in R$, where $X \in A$, we accept $r_{XY}$ with probability $w_{XY}$.
- If an arc $r_{XY}$ is accepted, we then add $Y$ to $A'$
- At the end of the process $A'$ contains a possible 1-hop augmentation of $A$

while it is analytically described in Algorithm 1. Using Algorithm 1 we can estimate the likelihood of a given set of beliefs (by dividing their observations by $n$) and infer a distribution of beliefs. Using any selection policy we wish, we can construct a set of likely beliefs we believe that the interlocutor knows. This policy can be either a specific cut-off, e.g. we only include an argument $A$ with $Pr(X) > 0.5$, or including each argument with probability $Pr(X)$. We denote the probability of accepting an arc as $Pr(r_{XY}) = w_{XY}$. We also assume the events of accepting $r_{ij}$ and $r_{km}$, where $r_{ij} \neq r_{km}$ to be independent but not mutually exclusive, therefore:

$$Pr(r_{ij} \cup r_{km}) = Pr(r_{ij}) + Pr(r_{km}) - Pr(r_{ij} \cap r_{km})$$

Appendix

Figure 3: (a) $RG_1$, (b), (c), (d) & (e) Possible augmentation $A'_R$, $A'_D$, $A'_F$, & $A'_{DF}$ respectively.
which means that the probability for a node to be added in \( A' \) follows Equation 5. Consequently the probability of obtaining a specific augmentation \( A' \) follows Equation 4, and therefore the described procedure essentially samples from the distribution of augmentations.

Assuming a sampling procedure which generates a number of \( n \) samples using the described method, then each node \( Y \) is included with probability equal to \( Pr(Y) \). Thus after \( n \) independent identically distributed (i.i.d.) samples, a proportion of \( nPr(Y) \) will contain argument \( Y \). We define \( k_Y = \sum_{i=1}^{n} I(Y \in A_i') \) where \( I \) is the indicator function taking the value of 1 if the predicate within is satisfied and \( A_i' \) is the set of nodes contained in the \( i \)-th sampled augmentation. In this case \( k_Y \) is used to denote the number of times we sampled \( Y \). The expected number of augmentations samples which contain \( Y \) follows a binomial distribution. Thus the expected number of observations \( k_Y \), of any given node \( Y \) after \( n \) tries is \( E(k_Y) = nPr(Y) \) and the variance is \( \text{Var}(k_Y) = nPr(Y)(1 - Pr(Y)) \). This defines a multinomial distribution over the set of nodes. Due to the law of large numbers and the fact that each sample is i.i.d., it holds that:

\[
E(k_Y) = nPr(Y) \\
\hat{P}r(Y) = \frac{k_Y}{n}
\]

where \( \hat{P}r(Y) \) denotes our estimates of \( Pr(Y) \). The procedure described here is a method to sample for the inclusion-exclusion probability. This is done due to the fact that the exhaustive calculation of this probability is exponential in the number of arcs considered.

**Algorithm 1 Monte-Carlo approximation**

**Input:** OneHopAugmentation\( (RG, \delta, \epsilon, A) \)

**Returns:** \( n, k_Y \in K \) for all \( Y \) which were sampled

\[
K \leftarrow \emptyset \\
n \leftarrow \frac{z_{1-\delta}}{4\epsilon^2} \\
\text{for } i = 1 \text{ to } n \text{ do}
\]

\[
A' \leftarrow A \\
\text{for all } r_{XY} \in R_A \text{ do}
\]

\[
\text{random} \leftarrow \text{Random}(0, 1) \\
\text{if } w_{XY} \geq \text{random then}
\]

\[
\text{if } Y \notin A' \text{ then}
\]

\[
A' \leftarrow A' \cup \{Y\} \\
\text{if } k_Y \in K \text{ then}
\]

\[
k_Y \leftarrow k_Y + 1 \\
\text{else}
\]

\[
k_Y \leftarrow 1 \\
K \leftarrow K \cup \{k_Y\}
\]

end if

end if

end for

end for

The generated random graphs are Poisson random graphs, with edge probability set to \( 50/n \) (where \( n \) is the size of the graph). The reason for selecting this probability was to ensure there exists a giant connected component and have a graph which is sufficiently dense to justify the need to use a sampling method to infer the argument likelihoods rather than to directly measure them, but also sufficiently sparse, in order to be able to use it on a computer with limited capabilities. Due to the lack of benchmarks to compare with, we do not know whether a random graph better corresponds to a realistic argument graph, however for the purposes of the Monte-Carlo simulation we believe that the graph structure is irrelevant for the purpose of argument likelihood estimation (assuming that the real argument graphs won’t be complete or have a path-like or grid-like structure).

4.1 Sampling accuracy & Experimental results

The number of samples \( n \) required to achieve an accuracy \( \epsilon \) with confidence \( \delta \) is given through the following theorem:

**Theorem 1** The Monte-Carlo approach to sample the probability distribution of the nodes in a set \( N_A \) is at least \( \epsilon \), close to convergence with probability at least \( \delta \) after:

\[
n = z_{1-\delta}^2 \frac{1}{4\epsilon^2} \text{ samples} \quad (7)
\]

**Proof 1 (Theorem 1)** Assume we sample \( n \) points from \( \mathcal{A}^N \). This would yield a sample \( K \). We now need to determine how far the empirical observations \( k_Y \) are from the expected \( E(k_Y) \). We use the normal approximation interval which can be found in various textbooks, e.g. [8]. Since we know that for each argument in \( N_A \) the probability to observe it in a given sample follows a binomial distribution, the normal approximation can be expressed as follows:

\[
Pr \left( |\hat{P}r(Y) - Pr(Y)| \geq z_{1-\delta} \frac{\sqrt{Pr(Y)(1-Pr(Y))}}{n} \right) \leq \delta \quad (8)
\]

where \( z_{1-\delta} \) is the normal distribution quantile function (the inverse of the CDF) which gives the value for which a standard normal random variable \( Y \) has the probability of exactly \( \delta \) to fall inside the range \( (-\infty, z_{1-\delta}) \). We set

\[
\epsilon = z_{1-\delta} \frac{\sqrt{Pr(Y)(1-Pr(Y))}}{n} \leq 0.5 \\
\text{which represents the desired upper bound for the error of our estimation. We are able to determine the accuracy of the estimation in relation to } n:
\]

\[
n = z_{1-\delta}^2 \frac{\hat{P}r(Y)(1 - \hat{P}r(Y))}{\epsilon^2} \quad (9)
\]

Since \( \hat{P}r(Y)(1 - \hat{P}r(Y)) \leq \frac{1}{4} \) then a general bound for \( n \) is:

\[
n \leq z_{1-\delta}^2 \frac{1}{4\epsilon^2} \quad (10)
\]
Equation 8 states our estimate $\hat{Pr}(Y)$ is $\epsilon$ close to the real value $Pr(Y)$ with probability less than $\delta$. In this case $\delta$ and $\epsilon$ are the accuracy and confidence parameters we require. Equation 9 gives us a strict bound on the rate of convergence of $\hat{Pr}(Y)$ to $Pr(Y)$. As a direct consequence we notice that values approaching $Pr(Y) = 0.5$ have the slowest convergence. On the other hand Equation 7 gives us a more generic with high probability bound for $n$.

In essence, Equation 7 gives us the expected upper bound of the error, which is expected to reach an accuracy equal to $\epsilon = 0.05$, with $\delta = 0.05$ confidence by taking $n \leq \frac{2.975}{\epsilon^2} \approx 385$ samples, independently of the size of the $RG$. Using the described algorithm we can estimate the likelihood of a given set of arguments (by dividing their observations by $n$) and infer a distribution of arguments. Based on these results we can augment our OM by choosing the set with the highest likelihood.

Performing tests on randomly generated graphs of various sizes we have obtained the results of the average error which can be seen in Figure 5. We can see that the average error is upper bounded by the theoretically expected value. In practice the convergence is much quicker than what the theory suggests, resulting in error less than 0.1 after only 15 samples. Additionally in Figure 4 we can see the error per argument likelihood $\hat{Pr}(Y)$. We observe how the error is maximized for likelihoods in the range of $0.4 - 0.6$ which further supports the results of Equation 9.

Using Spearman’s $\rho$ correlation coefficient, and assuming that we rank the variables with the variable ranked first being the variable with the highest likelihood, then Spearman’s $\rho$ would yield a value between 0 and 1. This value indicates how well the rankings of the theoretical values and the experimental values fit together, i.e. as $\rho$ tends closer to 1 then the rankings are more similar. There are several reasons to use Spearman’s $\rho$ over Pearson’s $r$ which are discussed in depth by N. Litvak et al [7].

In Figure 6 we can see the correlation of our samples with the real ranks of the nodes in $N_A$. Additionally we observe that the correlation $\rho \geq 0.95$ after around 80 samples at most, we should simply stop sampling since the improvement that would be yielded by additional samples would be of greatly diminished value. It is also worth noting that the graph size does not affect the accuracy of our estimations.

5 Related Work & Conclusions

Specifically, in the context of dialogue games, Riveret et al. [10; 11] model the possible knowledge of opponents in the form of arguments as we do. They nevertheless rely on the simplification that arguers are perfectly informed about all the arguments previously asserted by all their opponents in dialogues. Oren and Norman [9] propose a variant of the min-max algorithm for strategising through relying on models which represent both an agent’s knowledge, in the form of arguments, as well as their goals. However, nowhere in the aforementioned work is the problem of acquiring and maintaining, or furthermore augmenting an OM addressed. An exception, proposed by Black et al. [1], concerns a mechanism that enables agents to model preference information about others—which is important to another agent—and then rely on this information for making proposals that are more likely to be agreeable. In their case the mechanism responsible for developing a model of an agent’s preferences is explicitly provided, though they do not model agents’ knowledge. A similar approach to our work has been proposed Kovatsos et al. [12] who explored how to learn stereotypical sequences of utterances in dialogues for deducing an opponent’s strategy, though not relying on OMs. Finally, the work of Emel et al. [3] is worth noting, as it is similar to ours in the sense that they also explore the development of an OM but based on what norms or expectations an opponent might have, and not on its general beliefs.

Through this work we have provided a general methodology for updating and augmenting an OM, based on an agent’s experience obtained through dialogues. This methodology is based on two mechanisms respectively responsible for updating and augmenting an OM. In relation to the latter, we provided a method for building a graph between related arguments asserted by a modeller’s opponents, referred to a $RG$, and proposed an augmentation mechanism, enabling an agent to augment its current beliefs about its opponents beliefs by including additional information (arguments), that is of high likelihood to be related to what the opponent is currently assumed to know. Thus, we enabled an agent to also account...
in its strategising for the possibility that additional information may also be known to its opponents. Finally, we defined and analysed a Monte-Carlo simulation which enabled us to infer the likelihood of those additional arguments in a tractable and efficient way. We are aware that more investigation is needed with respect to relying on alternative contextual factors for quantifying the likelihood between elements in a \( RG \), such as the level of a participant’s membership in a group, and this is something we intend to investigate in the future. Finally, we also intend to evaluate the effectiveness of our approach when employed in accordance with trust related semantics [14] through which one may define the trustworthiness of an agent’s utterances with respect to its actual beliefs.

References