Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200519

An Argument-Based Framework for Selecting Dialogue Move Types and Content

Mark SNAITH [1]
Centre for Argument Technology, University of Dundee, UK

Abstract. Choosing what to say is an integral part of multi-party dialogue, whether to ensure a natural flow, or advance a strategy. This paper presents an argument-based framework for selecting dialogue move types and content. The framework first builds and evaluates arguments in favour of move types and content being preferred, before determining whether or not there is an *optimal* outcome - where both the move type and content are preferred - or a *sub-optimal* outcome - where either the type or content is preferred, but not both.

Keywords. dialogue, dialogue move selection, strategies, argumentation

1. Introduction

During the course of a dialogue, it is important that participants say the right thing at the right time. Formalised dialogue games (such as those specified by [1,2,3]), subsequently implemented as dialogue protocols (e.g. [4,5,6,7]), assist with this by mandating what types of move can follow others. This does however only partially address the problem: first, many dialogue games allow multiple valid move types at any given point, raising the question of exactly which move type to choose. Second, protocols do not influence the specific content of a move (i.e. the actual utterance made by the participant). While some protocols do place constraints on the content, this is only in an abstract sense; for instance, the content of a move must support a certain proposition p, but several different pieces of content might fulfil this criterion.

Dialogue move selection is a widely-studied topic, especially in the context of strategy [8,9,10]. Participants in a dialogue consider their strategic objectives before selecting a move type and appropriate content that they believe will stand the best chance of achieving those objectives.

Choosing what to say in a dialogue is fundamentally a decision problem. Such problems are well-suited to being solved using argumentation [11], with arguments being constructed for and against each possible alternative. Using argumentation to select dialogue moves and content was previously studied by [12], where the selection of a move type and content is a two-stage process: first by selecting a single move type based on *strategic goals* (the meta-level goals of the participant, such as “minimising the dialogue time”), then finding and selecting content to populate it based on *functional goals* (subject-specific goals, such as what the participant wants to achieve). If content cannot

[1]E-mail: m.snaith@dundee.ac.uk.

be found for the selected move type, it is assumed that the strategic and functional goals are incompatible.

A drawback of this approach is that it does not consider any alternative move types should no content be found. While this is justified in terms of a participant remaining true to their strategic goals, it assumes that a participant is always prepared to be bound by those goals even if it means they cannot speak in the dialogue. Considering the choice of content alongside the choice of move, and using the availability or otherwise of content to influence the set of available moves increases the chance of finding *something* to say. A participant then has the choice between saying this (while violating or ignoring some or all of their strategic goals), or saying nothing (and staying true to their strategic goals). This drawback was also identified by [13], who instead propose an algorithm that considers move types and content independently and concurrently, before choosing the "best" combination of type and content on the basis of these determinations.

Common to both [12] and [13] is that the determination is based on two sets of goals - strategic (relating to move types) and functional (relating to content). While considering goals is important, especially from a strategic point of view, limiting the determination to *only* considering goals is somewhat restrictive because there may be additional external factors that influence whether or not a move type and/or content to fill it is available. As an example, a software agent designed to recommend exercise might base its advice on the weather (i.e. outdoor vs indoor exercise), but the weather itself does not influence the agent's goals. Furthermore, an agent's goals might not be explicit, but rather determined by a variety of factors based on what the agent values in general and not just related to this current specific dialogue.

This paper presents an argument-based framework that selects between possible move types and content. As well as move types and associated content, the framework takes as further inputs: a set of properties that are assigned to move types and content; a set of values that move types and content can promote[2]; and a preference ordering over those values, determined by the current dialogue participant. The output is a move type and content selection that is either **optimal**, where both the type and content are more preferred with respect to other types and content, or **sub-optimal**, where either the type or content is preferred, but not both.

2. Preliminaries

2.1. Inputs to the framework

2.1.1. Move types and content

For the purposes of this work, a specific dialogue framework is not used, nor a specific protocol. Instead, the only assumption made is that a dialogue is taking place, at a certain point in that dialogue a set of move types M is available to a participant, and that there exists a set of content C that can be used to populate those move types; a content function that collects valid content for each move type is defined thus:

Definition 1 *Content:* $M \rightarrow 2^C$; $Content(m) = \{c \in C \mid c \text{ is valid content for } m\}$

[2] These values while similar in principle to those found in value-based argumentation [14] are not used in the same way in the present work.

$C_{Valid} = \bigcup_{m \in M} Content(m)$ represents all valid content across all move types.

2.1.2. Properties

Properties allow applications of the framework to consider external conditions that must be true for a certain move type and/or piece of content to be available. To again use the example of an agent recommending exercise, content that references walking a dog should only be considered if the person to whom the advice is being given has a pet dog. A *property function* is defined to assign properties to move types and content:

Definition 2 *Let P be a set of properties and $X \in \{M, C\}$, where M is the set of move types and C is the set of content: $Prop_X$: $X \rightarrow 2^P$; $Prop_X(\alpha) = \{p \mid \alpha \text{ has property } p\}$*

$Prop_D \subseteq P$ represents the set of properties that are currently true for the dialogue D.

2.1.3. Values and preferences

Values, and an associated preference ordering over them, are used to determine move types and content that are preferred. Values themselves are assigned to move types and content, while the preference ordering is individual to the dialogue participant. As with properties, a function is defined to assign values to move types and content.

Definition 3 *Let V be a set of values and $X \in \{M, C\}$, where M is the set of move types and C is the set of content: Val_X: $X \rightarrow 2^V$; $Val_X(\alpha) = \{v \mid v \text{ is promoted by } \alpha\}$*

Dialogue participants assign a (possibly partial) preference ordering, $<$, over values, which in turn will allow for a preference ordering over move types and content to be determined. This determination is defined in Section 3.

2.2. Argumentation - ASPIC+

ASPIC+ [15] is used as the basis the argumentation model. ASPIC+ combines the work of [16] with that of [17] to provide an account of structured argumentation from which an abstract argumentation framework [18] can be obtained and evaluated for acceptability. The three core elements of ASPIC+ are an *argumentation system*, a *knowledge base*, and an *argumentation theory*.

Definition 4 *An argumentation system is a tuple $AS = \langle \mathcal{L}, cf, \mathcal{R}, \leq \rangle$ where: $\mathcal{L}$ is a logical language; cf: $\mathcal{L} \rightarrow 2^{\mathcal{L}}$, a contrariness function; $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules; and $\leq$ is a partial preorder on $\mathcal{R}_d$.*

In the present work, a Prolog-style language for $\mathcal{L}$ is used, whose formal definition is left implicit, but informally contains terms that consist of: **atoms**, that begin with a lowercase character (e.g. x, y, $some\_string$); **variables**, that begin with an uppercase character (e.g. X, Y, $SomeVariable$); and **compound terms**, consisting of an atom and a (parameterised) list of variables and/or atoms (e.g. $term(SomeVariable)$, $another\_term(SomeVariable, some\_string)$). For brevity in presentation rules are also expressed in a Prolog-style such that a single rule defined over variables is provided in place of multiple concrete rules defined over atoms; for example, if $\mathcal{K} =$

$\{foo(term1), foo(term2)\}$, then $\mathcal{R}_d = \{foo(X) => bar(Y)\}$ is shorthand for $\mathcal{R}_d = \{foo(term1) => bar(term1), foo(term2) => bar(term2)\}$. The same notation is also used in defining contraries and preferences.

Definition 5 *A knowledge base in an argumentation system $AS = \langle \mathcal{L}, cf, \mathcal{R}, \leq \rangle$ is a pair $\langle \mathcal{K}, \leq' \rangle$, where $\mathcal{K} \subseteq \mathcal{L}$ and $\leq'$ is a partial preorder on $\mathcal{K} \setminus \mathcal{K}_n$.*

Definition 6 *An argumentation theory is a triple $AT = \langle AS, KB, \preccurlyeq \rangle$ where AS is an argumentation system, KB is a knowledge base in AS and $\preccurlyeq$ is an argument ordering on the set of all arguments that can be constructed from KB in AS.*

Argument orderings in an argumentation theory are used determine preferences, and subsequently defeat. In the present work, the weakest link principle is used to determine this ordering because it takes into account preferences over premises in arguments.

Definition 7 *Let A and B be two arguments. Then $A \prec B$ iff either:* ***(1)*** *B is firm and strict and A is defeasible or plausable; or* ***(2)*** *$LastDefRules(A) \preccurlyeq_S LastDefRules(B)$; or* ***(3)*** *$LastDefRules(A)$ and $LastDefRules(B)$ are empty and $Prem(A) \preccurlyeq_S Prem(B)$.*

In Definition 7, *LastDefRules* and *Prem* are functions that return, respectively, the last defeasible rules and premises of the given argument, and $\preccurlyeq_S$ denotes a set ordering.

3. Building the framework

Underpinning the framework is an Argumentation Theory, $AT = \langle AS, KB, \preccurlyeq \rangle$. The resultant argumentation framework from AT is evaluated under some semantics that is left open to specific applications. A sceptical semantics, such as grounded, will lead to an outcome only if the move type and content preferences resolve all conflicts. A credulous semantics, such as preferred, will reveal all mutually-exclusive available outcomes.

3.1. Knowledge base and preferences

The knowledge base is constructed on the basis of the available move types, content and their respective properties. In the remainder of this paper, the following abbreviated terms are used: *thp* means “type has property” *chp* means “content has property”; *icf* means “is content for”:

- $\forall c \in C_{Valid}$, $content(c) \in \mathcal{K}$; and $\forall p \in Prop_C(c)$, $chp(c,p) \in \mathcal{K}$, and if $p \in Prop_D$, $property(p) \in \mathcal{K}$, else $\neg property(p) \in \mathcal{K}$
- $\forall m \in M$, $type(m) \in \mathcal{K}$; and $\forall c \in Content(m)$, $icf(c,m) \in \mathcal{K}$
- $\forall m \in M$, $\forall p \in Prop_M(m)$: $thp(m,p) \in \mathcal{K}$; and if $p \in Prop_D$, $property(p) \in \mathcal{K}$, else $\neg property(p) \in \mathcal{K}$

If for some X, $\{t,c\}hp(X,p) \in \mathcal{K}$ and $property(p) \notin \mathcal{K}$, then $\neg property(P) \in \mathcal{K}$. This imposes a partial closure property on $\mathcal{K}$ in terms of properties: if properties we know can exist (from the $\{t,c\}hp$ terms) are not explicitly true, they are assumed false.

Knowledge base preferences are determined from the preferences over the values they promote. As well as considering preferences over move types and content separately, preferences between move types and content are also permitted, i.e. a piece of content can be preferred to a certain move type. This allows the framework to resolve conflict between possible sub-optimal outcomes by considering whether or not a preferred move type is further preferred to a preferred piece of content, or vice versa.

Generating the knowledge base preferences requires a determination of preference over sets of values rather than individual values themselves. To do this, the *democratic determination* [19] is used:

$\forall c_1, c_2 \in C_{Available}$ s.t. $c_1 \neq c_2$: **(1)** if $Val_C(c_1) = \emptyset$ then $content(c_1) \not\leq content(c_2)$; else **(2)** if $Val_C(c_2) = \emptyset$ and $Val_C(c_1) \neq \emptyset$ then $content(c_1) \leq content(c_2)$; else **(3)** $content(c_1) \leq content(c_2)$ if $\forall X \in Val_C(c_1)$, $\exists Y \in Val_C(c_2)$ s.t. $X \leq Y$

$\forall m_1, m_2 \in M$ s.t. $m_1 \neq m_2$: **(1)** if $Val_M(m_1) = \emptyset$ then $type(m_1) \not\leq type(m_2)$; else **(2)** if $Val_M(m_2) = \emptyset$ and $Val_M(m_1) \neq \emptyset$ then $type(m_1) \leq type(m_2)$; else **(3)** $type(m_1) \leq type(m_2)$ if $\forall X \in Val_M(m_1)$, $\exists Y \in Val_M(m_2)$ s.t. $X \leq Y$

$\forall m \in M$, $\forall c \in C$: **(1)** if $Val_M(m) = \emptyset$ then $type(m) \not\leq content(c)$; else **(2)** if $Val_C(c) = \emptyset$ and $Val_M(m) \neq \emptyset$ then $type(m) \leq content(c)$; else **(3)** $type(m) \leq content(c)$ if $\forall X \in Val_M(m)$, $\exists Y \in Val_C(c)$ s.t. $X \leq Y$

3.2. Contrariness

Identifying the most preferred move types and content is a binary problem, insofar as if one move type (resp. piece of content) is most preferred, no others can be. To model this, we declare that *pt* ("preferred type") and *pc* ("preferred content") terms are contraries of themselves, except where specific instantiations would self-attack. In terms of outcomes, an optimal outcome (*opt*) should always attack a sub-optimal (*sub_opt*) outcome, while all sub-optimal outcomes should be in conflict with each other, again except where their arguments are assigned to the same atoms. Formally:

- $Cf(pt(Type1)) = \{pt(Type2)\}$, where $Type1 \neq Type2$;
- $Cf(pc(Content1)) = \{pc(Content2)\}$, where $Content1 \neq Content2$;
- $Cf(opt(\_,\_)) = \{sub\_opt(\_,\_)\}$, where _ represents any atom;
- $Cf(sub\_opt(Type1, Content1) = \{sub\_opt(Type2, Content2)\}$, where if $Type1 = Type2$, $Content1 \neq Content2$ and if $Content1 = Content2$, $Type1 \neq Type2$

3.3. Rules

One of the strengths of the framework is that move types and content are considered separately, before being brought together to determine outcomes. This achieved through the rules in AS, which are $\mathcal{R}_d = \{r_1, r_2, r_4, r_5, r_7, r_8, r_9\}$ and $\mathcal{R}_s = \{r_3, r_6\}$, where:

r_1 : $property(Property), thp(MoveType, Property) \Rightarrow at(MoveType)$
r_2 : $at(MoveType), type(Type) \Rightarrow pt(MoveType)$
r_3 : $\neg property(Property)\ thp(MoveType, Property) \rightarrow \neg at(MoveType)$
r_4 : $property(Property), chp(Content, Property) \Rightarrow ac(Content)$
r_5 : $ac(Content)\ content(Content) \Rightarrow pc(Content)$
r_6 : $\neg property(Property), chp(Content, Property) \rightarrow \neg ac(Content)$
r_7 : $pt(MoveType), pc(Content), icf(Content, MoveType) \Rightarrow opt(MoveType, Content)$

r_8 : $at(MoveType), pc(Content), icf(Content, MoveType) \Rightarrow sub\_opt(MoveType, Content)$
r_9 : $pt(MoveType), ac(Content), icf(Content, MoveType) \Rightarrow sub\_opt(MoveType, Content)$

Rules r_1 and r_3, and r_5 and r_6 determine whether or not move types (resp. content) are available based on their properties. Rules r_2 and r_5 create arguments for move types (resp. content) being preferred. Rules r_7 through r_9 determine outcomes, either optimal (r_7) or two types of sub-optimal: incorporating a preferred move type (r_8), or preferred content (r_9). Also, two preferences over rules are defined: $r_8 < r_7$ and $r_9 < r_7$, ensuring that when there is an optimal outcome, all sub-optimal outcomes are defeated.

4. Examples

Here, three concrete examples are presented that illustrate applications of the framework using different preference orderings over values. All examples use the following knowledge base $\mathcal{K}$, whose construction is left implicit, and chp, thp and icf have the same meaning as in Section 3:

$$\left\{\begin{array}{l} property(p),\ \neg property(q),\ content(\phi),\ chp(\phi,p), chp(\psi,p), \\ content(\theta),\ chp(\theta,q),\ type(assert),\ thp(assert,p),\ type(question), \\ thp(question,p),\ icf(\psi,assert),\ icf(\phi,question),\ icf(\theta,assert) \end{array}\right\}$$

Notice that $\neg property(q), chp(\theta,q) \in K$. This means that, as a result of the strict rule $r6$ (defined in section 3), any argument for $ac(\theta)$ is strictly defeated, as are any other arguments in which it is a sub-argument. Since all rules (and thus all arguments) for optimal and sub-optimal outcomes rely on content being available, θ is not considered in any of the examples, illustrating the impact properties have in the framework.

Values and properties are assigned to move types and content as follows:

$Val_M(assert) = \{v_1\}$
$Val_M(question) = \{v_2\}$
$Val_C(\phi) = \{v_3\}$
$Val_C(\psi) = \{v_4\}$
$Val_C(\theta) = \{v_5\}$
$Prop_M(assert) = \{p\}$
$Prop_M(question) = \{p\}$
$Prop_C(\phi) = \{p\}$
$Prop_C(\psi) = \{p\}$
$Prop_C(\theta) = \{q\}$

4.1. Example 1: optimal outcome

Assume that the value preferences are: $v_4 < v_3 < v_2 < v_1$. On the basis of these preferences, we obtain a preference ordering over $\mathcal{K}$: $content(\psi) < content(\phi) < type(question) < type(assert)$. From $\mathcal{K}$ and the preferences over $\mathcal{K}$, the resultant arguments and defeats yield an argumentation framework that is shown (for clarity only partially) on the left of Figure 1 after evaluation under grounded semantics, with a solid line indicating "acceptable" and a dashed line indicating "unacceptable". The argument labels correspond to the following conclusions:

$A12$: $pc(\phi)$
$A13$: $pt(question)$
$A16$: $pt(assert)$
$A18$: $sub\_opt(assert,\phi)$
$A19$: $opt(assert,\phi)$
$A21$: $pc(\psi)$
$A22$: $opt(question,\psi)$
$A23$: $sub\_opt(question,\psi)$
$A24$: $sub\_opt(question,\psi)$
$A25$: $sub\_opt(assert,\phi)$

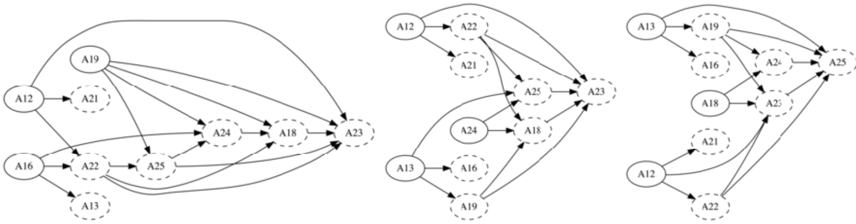

Figure 1. Partial argumentation frameworks for Examples 1 (L), 2 (C) and 3 (R)

Argument $A19$ is acceptable, representing an **optimal outcome**, of move type *assert* and content ϕ.

4.2. Example 2: sub-optimal outcome with preferred move type

Assume that the value preferences are changed to: $v_4 < v_3 < v_1 < v_2$. These lead to the knowledge base preferences: $content(\psi) < content(\phi) < type(assert) < type(question)$, which yield the argumentation framework partially shown in the centre of Figure 1, where the argument labels are the same as in Example 1. Argument $A24$ is acceptable, representing a **sub-optimal outcome** of move type *question* with content ψ. This is consistent with the preferences: *question* is more preferred to *assert*, but ψ is less preferred to ϕ, thus we could only have a sub-optimal outcome; the chosen sub-optimal outcome arises because *question* is more preferred to ϕ.

4.3. Example 3: sub-optimal outcome with preferred content

This final example uses value preferences: $v_1 < v_4 < v_2 < v_3$. These lead to the knowledge base preferences: $type(assert) < content(\psi) < type(question) < content(\phi)$, which yield the argumentation framework partially shown in the right of Figure 1, where again the argument labels are the same as in Example 1. Argument $A18$ is acceptable, representing a **sub-optimal outcome** of move type *assert* with content ϕ. This too is consistent with the preferences: ϕ is more preferred to ψ, but *assert* is less preferred to *question*, thus we could only have a sub-optimal outcome; the chosen sub-optimal outcome arises because ϕ is more preferred to *question*.

5. Summary and conclusions

This paper has presented an argument-based framework for selecting dialogue move types and content. The framework takes into account necessary properties, values promoted by move types and content, and a preference ordering over those values by the dialogue participant. By constructing arguments in favour of preferred move types and content, the framework can determine *optimal* and *sub-optimal* outcomes: an optimal outcome is where a preferred move type can be matched with preferred content; a sub-optimal outcome is where only the move type or the content is preferred.

Directions for future work include examining the properties of the framework to determine whether or not an outcome (whether optimal or sub-optimal) can always be reached. Additionally, the framework could be extended to take into account exactly how the set of possible content is arrived at for each move type. If for instance pieces of content are themselves the conclusions of arguments, those arguments may influence the values the content promotes, or the preference ordering over those values. Further extensions will also examine refinement and/or expansion of the argument model to either increase the possibility of yielding only a single outcome (optimal or sub-optimal), or providing an additional step to further choose between them.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553.

References

[1] Hamblin CL. Fallacies. Methuen & Company; 1970.
[2] Walton DN. Logical Dialogue-Games and Fallacies. Lanham: University Press of America; 1984.
[3] Walton DN, Krabbe ECW. Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, New York; 1995.
[4] McBurney P, Parsons S. Dialogue Games in Multi-Agent Systems. Informal Logic. 2002;22(3):257–274.
[5] McBurney P, Parsons S. Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. Journal of Logic, Language and Information. 2002;11:315–334.
[6] Prakken H. Formal systems for persuasion dialogue. The knowledge engineering review. 2006;21(2):163–188.
[7] Black E, Hunter A. An inquiry dialogue system. Autonomous Agents and Multi-Agent Systems. 2009;19:173–209.
[8] Kakas A, Maudet N, Moraitis P. Layered strategies and protocols for argumentation-based agent interaction. In: International Workshop on Argumentation in Multi-Agent Systems. Springer; 2004. p. 64–77.
[9] Oren N, Norman T, Preece A. Loose Lips Sink Ships: a Heuristic for Argumentation. In: Maudet N, Parsons S, Rahwan I, editors. Proceeings of the Third International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006); 2006. .
[10] Parsons S, McBurney P, Sklar E, Wooldridge M. On the relevance of utterances in formal inter-agent dialogues. In: International Workshop on Argumentation in Multi-Agent Systems. Springer; 2007. p. 47–62.
[11] Amgoud L, Prade H. Using arguments for making and explaining decisions. Artificial Intelligence. 2009;173(3-4):413–436.
[12] Amgoud L, Hameurlain N. An argumentation-based approach for dialog move selection. In: Maudet N, Parsons S, Rahwan I, editors. Proceeings of the Third International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006). Springer; 2006. p. 128–141.
[13] Beigi A, Mozayani N. A new dialogue strategy in multi-agent systems. Journal of Intelligent and Fuzzy Systems. 2014 01;27:641–653.
[14] Bench-Capon TJM. Value-based argumentation frameworks. In: Proceedings of Non Monotonic Reasoning; 2002. p. 444–453.
[15] Prakken H. An abstract framework for argumentation with structured arguments. Argument and Computation. 2010;1(2):93–124.
[16] Pollock JL. Defeasible Reasoning. Cognitive Science. 1987;11:481–518.
[17] Vreeswijk GA. Abstract Argumentation Systems. Artificial Intelligence. 1997;90:225–279.
[18] Dung PM. On the acceptability of arguments and its fundemental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77:321–357.
[19] Caminada M, Modgil S, Oren N. Preferences and unrestricted rebut. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Proceddings of the Fifth International Conference on Computational Models of Argument (COMMA 2014). IOS Press; 2014. p. 209–220.