Acceptance Trees for Dialogue and Decision Support over Bipolar Argumentation Frameworks

Stefano Bistarelli[1], Marco Cuccarini[1,2] and Carlo Taticchi[1,*]

[1]University of Perugia, Italy
[2]University of Naples Federico II, Italy

Abstract

We introduce Acceptance Trees (ATs), tree-shaped explanatory structures aimed at supporting interpretation and interactive reasoning in argumentation-based systems. Instead of presenting the full graph to the user, ATs are extracted from Bipolar Argumentation Frameworks (BAFs), which provide the underlying argumentative structure. The resulting trees guarantee the justification of a designated root argument, while offering a more intuitive and navigable structure for explanation and dialogue. We provide a formal definition of ATs, analyse their structural and semantic properties, and present a general construction method applicable to arbitrary BAFs, together with an executable encoding in Answer Set Programming. Finally, we show how ATs can be integrated into an argumentation-based dialogue procedure for decision support and persuasion, where they regulate argument selection and disclosure, supporting transparent and explainable interaction grounded in formal argumentation.

Keywords

Computational Argumentation, Bipolar Argumentation Frameworks, Automated Reasoning, Explainable Decision Making

1. Introduction

Computational Argumentation provides a principled framework for modelling reasoning and debate by explicitly representing claims together with the attack and support relations that determine their acceptability [1]. Its formal foundations enable transparent and explainable reasoning, and a growing body of work has explored how argumentation can be embedded into interactive settings, ranging from explanation tools to conversational agents [2, 3].

However, integrating argumentation into genuinely dialogical environments introduces challenges that go beyond the static reasoning tasks typically studied in the literature. This is well illustrated by the International Competition on Computational Models of Argumentation (ICCMA) [4, 5, 6, 7], where the main tasks, such as computing extensions or deciding credulous and sceptical acceptance, assume a fully specified argumentation framework as input, with all arguments and relations given in advance and evaluated globally in a single step. Although recent dynamic tracks consider sequences of incremental modifications [6, 7], the emphasis remains on efficiently recomputing acceptability under structural updates rather than on how an existing, fixed knowledge base (KB) should be progressively disclosed to a user.

In a dialogue setting, by contrast, the argumentative KB is gradually revealed through interaction. This raises the question of how to ensure that, at each step of the conversation, the portion of the knowledge disclosed so far remains internally consistent and continues to support or challenge a designated claim, rather than relying on properties that only hold at the level of the fully specified framework.

Our goal is to provide a formal basis for natural-language conversations in which a user discusses a given topic through an argumentation-based dialogue procedure. Rather than allowing arguments to be generated freely, the procedure assumes a predefined argumentative KB and selects responses

SAFA'26: Sixth International Workshop on Systems and Algorithms for Formal Argumentation, September, 2026, Barcelona, Spain
*Corresponding author.
stefano.bistarelli@unipg.it (S. Bistarelli); marco.cuccarini@unina.it (M. Cuccarini); carlo.taticchi@unipg.it (C. Taticchi)
0000-0001-7411-9678 (S. Bistarelli); 0009-0007-8943-8090 (M. Cuccarini); 0000-0003-1260-4672 (C. Taticchi)

by progressively disclosing arguments that are logically justified with respect to the current state of the dialogue. This requires a representation that not only captures the underlying argumentative knowledge, but also supports its incremental presentation while preserving the correctness of the reasoning throughout the interaction.

This paper addresses these issues by combining Bipolar Argumentation Frameworks (BAFs) [8] with a dialogue-oriented perspective. We adopt BAFs as the underlying representation due to their ability to jointly capture attack and support relations among arguments. Building on this representation, we introduce Acceptance Trees (ATs), tree-shaped explanatory structures extracted from a BAF that capture a line of reasoning leading to the acceptance of a designated root argument under the adopted admissible semantics. Rather than serving as a mere reformulation of the underlying graph, ATs are designed to support incremental disclosure by construction: arguments can be revealed progressively while every intermediate view preserves a coherent justification for the root.

We then illustrate how ATs can underpin an argumentation-based dialogue procedure for decision support and persuasion. Within this procedure, the conversation unfolds through the progressive disclosure of arguments from an AT, according to the user's contributions and the selected interaction objective. As a result, explainability is preserved throughout the dialogue, rather than reconstructed only after the reasoning process has concluded.

The remainder of the paper is organised as follows. Section 2 recalls the necessary background on BAFs and the notion of defence. Section 3 introduces ATs, studies their properties, and discusses their construction from BAFs. Section 4 presents the dialogue procedure and describes how ATs are exploited to generate responses during the interaction. Section 5 reviews the most closely related work. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Background

Computational Argumentation is an interdisciplinary field that combines Computer Science, Logic and Artificial Intelligence to model, analyse and evaluate arguments in automated systems. It involves the formalisation of arguments and reasoning processes designed to identify and evaluate conflicting positions. A key concept in Computational Argumentation is that of Abstract Argumentation Frameworks (AFs) [9], which provide formal structures for representing and reasoning about arguments and attacks between them. AFs are highly versatile but quite basic in their expressiveness, and as such, they have been extended in various ways to capture different nuances of human reasoning. Bipolar Argumentation Frameworks (BAFs) [8] are an extension of classical AFs that incorporate support and attack relations between arguments.

Definition 1. *A Bipolar Argumentation Framework (BAF) is a tuple $\langle A, Att, Supp \rangle$ where A is a finite set of arguments, $Att \subseteq A \times A$ is the attack relation, where $(a, b) \in Att$ means that argument a attacks argument b, and $Supp \subseteq A \times A$ is the support relation, where $(a, b) \in Supp$ means that argument a supports argument b.*

The acceptability of arguments is then evaluated using argumentation semantics [1], i.e. rules designed to determine which arguments are justified or rejected based on their relations with other arguments. Acceptability often involves considering a notion of conflict-freeness, i.e., two arguments that are acceptable together must not attack each other. Additionally, for an argument to be accepted in a BAF, it typically needs to be defended, which means that for every attack directed at it, there must be a counter-argument that can neutralise that attack.

The role of support relations in BAFs can be interpreted in different ways, as discussed in the literature [10]. These interpretations differ in how support affects the acceptability of arguments and in the kinds of interactions it induces with attacks. In this work, we adopt the *necessary* interpretation of support: if an argument a supports b, then b is acceptable only if also a is acceptable. Necessary support is particularly well suited to explanation and persuasion, since it captures a dependency notion in which claims are justified only if their underlying reasons are themselves valid. The necessary reading of support induces additional forms of attack, called *extended* attacks [10], which make explicit how attacks can propagate through support relations.

Definition 2. *Let $B = \langle A, Att, Supp \rangle$ be a BAF. There is an* extended attack *from a to b if and only if one of the following conditions holds:*

1. $(a, b) \in Att$*;*
2. *there exists a sequence of arguments $a_1, a_2, \ldots, a_n$ with $n \geq 3$ such that $(a_1, a_2) \in Att$, $(a_2, a_3) \in Supp, \ldots, (a_{n-1}, a_n) \in Supp$, with $a_1 = a$ and $a_n = b$;*
3. *there exists a sequence of arguments $a_1, \ldots, a_n$ with $n \geq 2$ and some a_p such that $(a_1, a_2) \in Supp, \ldots, (a_{n-1}, a_n) \in Supp$ and $(a_1, a_p) \in Att$, with $a_n = a$ and $a_p = b$.*

Figure 1 illustrates how the direct attack from a to b propagates through the support relations. Since b supports c, which in turn supports d, a also attacks c and d. At the same time, since a supports e, which supports f, both e and f also attack b.

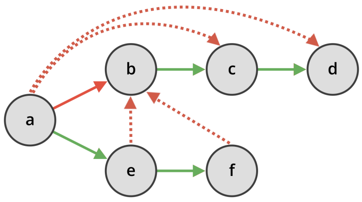

Figure 1: Example of a BAF where green edges represent supports and red ones attacks. Dotted edges are extended attacks.

At this point, we can characterise when an argument is defended against all its (direct and extended) attackers.

Definition 3. *Let $B = \langle A, Att, Supp \rangle$ be a BAF, and let Att^* denote the set of extended attacks induced by B. Given a set of arguments $S \subseteq A$, we say that S* defends *an argument $a \in A$ if, for every argument $b \in A$ such that $(b, a) \in Att^*$, there exists an argument $c \in S$ such that $(c, b) \in Att^*$.*

The notion of defence naturally gives rise to admissibility. In particular, we adopt the labelling-based characterisation of admissible semantics for BAFs, where arguments labelled IN are accepted, arguments labelled OUT are rejected, and the remaining ones are labelled UNDEC.

Definition 4. *Let $B = \langle A, Att, Supp \rangle$ be a BAF, and let Att^* denote the set of extended attacks induced by B. An* admissible labelling *of B is a function $\mathcal{L}^B_{adm} : A \rightarrow \{\text{IN}, \text{OUT}, \text{UNDEC}\}$ such that, for every argument $a \in A$:*

- $\mathcal{L}^B_{adm}(a) = \text{IN}$ *if for all $b \in A$ with $(b, a) \in Att^*$, it holds that $\mathcal{L}^B_{adm}(b) = \text{OUT}$;*
- $\mathcal{L}^B_{adm}(a) = \text{OUT}$ *if and only if there exists $b \in A$ such that $(b, a) \in Att^*$ and $\mathcal{L}^B_{adm}(b) = \text{IN}$;*
- $\mathcal{L}^B_{adm}(a) = \text{UNDEC}$ *otherwise.*

3. Acceptance Trees

While BAFs provide a rich representational formalism, their graph structure is not well suited for interactive explanation. In a dialogue setting, arguments need to be presented incrementally, in a way that remains intelligible without requiring access to the full framework at each step. However, BAFs may contain cycles and mutually dependent attack and support relations, which prevents a natural decomposition into a linear sequence of independent reasoning steps. Moreover, argumentation semantics are defined over the entire framework and determine only the final acceptance status of arguments, without providing a structured form of justification that can directly guide stepwise presentation to a user.

We model a discussion around a main claim by representing it as a BAF rooted at the corresponding argument, with the remaining arguments capturing supporting and opposing reasons and their interrelations. In this setting, the objective is to ensure that the root argument is defended against all its direct and extended attackers, while the attacks and supports directed towards it constitute its underlying justification. To bridge the gap between the possibly cyclic structure of a BAF and the need for a structured, stepwise explanation, we introduce the Acceptance Tree, a tree-shaped substructure of the underlying BAF rooted at the main claim. It is obtained by selecting a subset of arguments and relations that i) preserve the defence of the root, ii) ensure that every argument is connected to it via a directed path, and iii) restrict each argument to have at most one outgoing edge.

Definition 5. *Let $B = \langle A, Att, Supp \rangle$ be a BAF. An* Acceptance Tree *(AT) for an argument $r \in A$ is a tuple $T = \langle A_T, Att_T, Supp_T \rangle$ such that:*

- $A_T \subseteq A$*,* $Att_T \subseteq Att \cap (A_T \times A_T)$*, and* $Supp_T \subseteq Supp \cap (A_T \times A_T)$*;*
- *there exists an admissible labelling* $\mathcal{L}^T_{adm}$ *of* T *such that* $\mathcal{L}^T_{adm}(r) = \text{IN}$*;*
- $(r, a) \notin Att_T \cup Supp_T$ *for every* $a \in A_T$*;*
- *for each* $x \in A_T \setminus \{r\}$*, there is at most one* $y \in A_T$ *such that* $(x, y) \in Att_T \cup Supp_T$*;*
- *for each* $a \in A_T \setminus \{r\}$*, there exists a directed path from* a *to* r *in* T*.*

Figure 2 shows an AT rooted at d, with $Att_T = \{(b, c), (e, b)\}$ and $Supp_T = \{(a, b), (c, d)\}$. Every non-root argument has at most one outgoing relation and a directed path to d, while the root has no outgoing relations. Moreover, the labelling that assigns IN to a, c, d, and e, and OUT to b, is admissible. Therefore, the structure satisfies all the conditions of Definition 5.

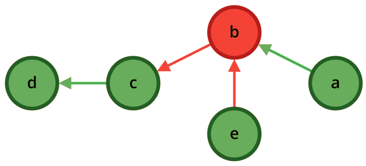

Figure 2: Example of an AT rooted at d. Green arguments are labelled IN, while the red argument is labelled OUT.

An AT defending a main claim can be systematically constructed starting from a BAF and a designated root argument. The construction relies exclusively on the information encoded in the BAF, namely its arguments together with their attack and support relations, and does not require the introduction of external information or additional reasoning principles. In the following discussion, we outline a construction procedure and provide an executable encoding based on Answer Set Programming (ASP) [11], together with a correctness result.

3.1. Construction of Acceptance Trees

We provide an ASP-based approach to construct an AT from a given BAF with respect to a designated root argument. The method systematically generates tree-shaped substructures rooted at the target argument, subject to structural constraints ensuring connectivity and acyclicity, and selects those substructures that preserve the attack and support relations relevant for the defence of the root. In particular, all attacks against the root must be countered within the resulting structure, and every included argument is required to contribute, directly or indirectly, to its defence. This guarantees that the extracted AT provides a coherent justification for the root argument within the original BAF.

Listing 1 encodes the initial step of the construction. Initially, the input to the program must specify the set of arguments (`arg/1`), the attack relations (`att/2`), the support relations (`supp/2`), and the designated root (`root/1`). The root argument is always included in the tree via the rule `arg_tree(R) :- arg(R), root(R)`. Other arguments are then selected non-deterministically with the choice rule `arg_tree(X) : arg(X)` , allowing multiple combinations of arguments to be considered. Similarly, edges representing attacks and supports are non-deterministically included among the selected arguments: `att_tree(X,Y)` is added if X and Y are in the tree and `att(X,Y)` exists in the original BAF, while `supp_tree(X,Y)` is added analogously for supports. Finally, the auxiliary predicate `edge_tree(X,Y)` collects all edges in the tree, whether attacks or supports.

Listing 1: Program initialisation.

```
arg_tree(R) :- arg(R), root(R).
{ arg_tree(X) : arg(X) }.
{ att_tree(X,Y) : arg_tree(X), arg_tree(Y), att(X,Y) }.
{ supp_tree(X,Y) : arg_tree(X), arg_tree(Y), supp(X,Y) }.
edge_tree(X,Y) :- att_tree(X,Y).
edge_tree(X,Y) :- supp_tree(X,Y).
```

The constraints in Listing 2 enforce structural properties on the candidate tree to ensure it forms a proper rooted subtree. The first constraint, `:- edge_tree(R,_), root(R)`, guarantees that the root argument has no outgoing edges, preserving its role as the origin of the tree. The second constraint, `:- edge_tree(Z,X), edge_tree(Z,Y), X != Y`, ensures that each non-root argument has at most one outgoing edge, preventing branching at non-root nodes. Reachability from the root is defined by the recursive rules `reach(R) :- root(R)` and `reach(Y) :- edge_tree(Y,X), reach(X)`, which propagate the reachability predicate along the edges of the tree. Finally, the constraint `:- arg_tree(X), not reach(X)` ensures that every argument selected in the tree is connected to the root, discarding any disconnected nodes and guaranteeing the resulting structure is a single connected subtree.

Listing 2: Structural constraints

```
:- edge_tree(R,_), root(R).
:- edge_tree(Z,X), edge_tree(Z,Y), X != Y.
reach(R) :- root(R).
reach(Y) :- edge_tree(Y,X), reach(X).
:- arg_tree(X), not reach(X).
```

The extended attack relation induced by necessary support of [10] within the constructed tree is encoded using the predicate `attack/2` in Listing 3. The rule `supp_tree(X,Z) :- supp_tree(X,Y), supp_tree(Y,Z)` ensures that support is transitive, propagating support along chains of arguments within the tree. Then, `attack(X,Y) :- att_tree(X,Y)` captures direct attacks, corresponding to explicit edges in the original BAF, while rules `attack(X,Y) :- att_tree(Z,Y), supp_tree(Z,X)` and `attack(X,Y) :- att_tree(X,Z), supp_tree(Z,Y)` encode extended attacks from Definition 2. The predicate `attacked/1` identifies those arguments that are attacked within the current tree by already accepted arguments, thereby providing the basis for the definition of defence.

Listing 3: Necessary support.

```
supp_tree(X,Z) :- supp_tree(X,Y), supp_tree(Y,Z).
attack(X,Y) :- att_tree(X,Y).
attack(X,Y) :- att_tree(Z,Y), supp_tree(Z,X).
attack(X,Y) :- att_tree(X,Z), supp_tree(Z,Y).
attacked(X) :- attack(Y,X), in(Y).
```

The encoding of an admissible set within the tree, shown in Listing 4, adapts the approach originally proposed in [12] from AFs to the bipolar setting. The first two rules (`in/1`, `out/1`) non-deterministically guess a candidate admissible set by assigning each argument either to `in` or to `out`. Conflict-freeness is enforced by the integrity constraint `:- in(X), in(Y), attack(X,Y)`, which prohibits the simultaneous acceptance of arguments that attack each other. The notion of defence is encoded by requiring every attacker of an accepted argument to be counterattacked by another accepted argument, thus realising a collective notion of defence over the tree structure. Operationally, the predicate `not_defended/1` identifies arguments for which this condition is violated, and the constraint `:-`

`in(X), not_defended(X)` ensures that only defended arguments can belong to the admissible set. Finally, the rule `:- root(R), not in(R)` requires the designated root argument to belong to the admissible set, thereby ensuring that it is defended in the resulting tree.

Listing 4: Defence and enforcement of the root.

```
in(X) :- arg(X), not out(X).
out(X) :- arg(X), not in(X).
not_defended(X) :- attack(Y,X), not attacked(Y).
:- in(X), in(Y), attack(X,Y). %conflict-free
:- in(X), not_defended(X). %collective defense
:- root(R), not in(R).
```

Proposition 1. *Let $B = \langle A, Att, Supp \rangle$ be a BAF and $r \in A$ be a designated root argument. Then, any tuple $T = \langle A_T, Att_T, Supp_T \rangle$ generated by the ASP program described in Listings 1–4, through the predicates `arg_tree/1`, `att_tree/2`, and `supp_tree/2`, constitutes an AT for r.*

Proof. The correctness of the construction follows directly from the rules and constraints encoded in the ASP program. All arguments, attacks, and supports in T are non-deterministically selected from B via the predicates `arg_tree/1`, `att_tree/2`, and `supp_tree/2` in Listing 1, ensuring that T is a subframework of B. From [12] and the enforcement constraint `:- root(R), not in(R)` in Listing 4 we have that the root r is labelled IN under the admissible semantics computed on T.

We now show that T forms a directed tree rooted at r. First, the constraint `:- edge_tree(R,_), root(R)` in Listing 2 ensures that the root has no outgoing edges. Next, `:- edge_tree(Z,X), edge_tree(Z,Y), X != Y` guarantees that each non-root argument has at most one outgoing edge in T. Finally, the reachability rules `reach/1` and the associated constraint `:- arg_tree(X), not reach(X)` in Listing 2 ensure that every argument included in A_T is connected to the root. □

Consider the BAF shown in Figure 3. In this framework, the argument d is not defended against the extended attacks induced by the arguments in the cycle involving b, c, and e. Nevertheless, it is possible to extract an AT rooted at d by selecting the arguments $\{a, b, c, d, e\}$ together with the support relations (a, b) and (c, d), and the attack relations (b, c) and (e, b), while discarding the attack (c, e), which is involved in the cycle and does not contribute to the defence of d. The resulting structure, shown in Figure 2, is acyclic and its root argument d is labelled IN under the admissible semantics. Running the ASP encoding with clingo on this instance yields four distinct AT structures. The one shown in Figure 2 is the largest in terms of number of arguments.

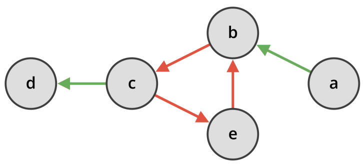

Figure 3: Example of BAF containing a cycle. Argument d is not acceptable under admissible semantics.

3.2. Canonical Acceptance Tree

A trivial solution to the construction problem always exists and consists of the AT containing only the root argument. Since no additional arguments or relations are selected, all structural constraints are trivially satisfied and the root is labelled IN by construction. Although always valid, such a solution provides virtually no explanatory information. Although explanatory quality is application-dependent, the amount of argumentative structure retained from the original BAF can be used as a structural proxy for informativeness. The construction of an AT is inherently non-deterministic. Given a BAF and a designated root argument, multiple ATs may satisfy Definition 5, each representing a different justification for the acceptance of the root. This multiplicity can be exploited to tailor the extracted explanation to different application requirements.

A natural way of selecting among alternative ATs is to formulate the extraction process as an optimisation problem. Let $\mathcal{T}(B, r)$ denote the set of all ATs for the root argument r of a BAF B. Given an objective function $f : \mathcal{T}(B, r) \to \mathbb{R}$, a canonical AT can be selected as

$$T^{\star} \in \arg \max_{T \in \mathcal{T}(B,r)} f(T),$$

or equivalently by minimisation, depending on the chosen objective.

Different objective functions induce different notions of canonicality. For instance, following the principle of minimal change advocated in enforcement approaches for AFs [13], one may favour ATs that retain as much information as possible from the original BAF. Conversely, minimisation can be used to obtain more concise ATs when combined with an appropriate non-triviality or coverage requirement. Different weights can also be assigned to attacks and supports, depending on the intended explanatory emphasis.

As a concrete refinement, we consider the number of retained arguments as a structural measure of informativeness and select an AT containing the maximum possible number of arguments. This criterion is implemented by the optimisation statement in Listing 5.

Listing 5: Maximisation of the number of arguments in an AT.

```
#maximize { 1@1,X : arg_tree(X) }.
```

Under this refinement, the singleton AT is returned only when no AT containing additional arguments exists. Analogous optimisation statements over the predicates representing selected attacks and supports can be used to implement the other criteria described above. The ASP program guarantees that every optimal solution still satisfies Definition 5 and is optimal with respect to the selected criterion. We remark that these optimisations provide a precise structural notion of informativeness, rather than an absolute guarantee of explanatory quality, which may also depend on the application context and the user.

The optimisation perspective also provides a natural foundation for comparing different ATs semantically. While classical extension-based semantics only distinguish whether the root is accepted or not, graded and ranking semantics for bipolar argumentation assign a degree of acceptability or strength to each argument [14, 15, 16]. Such semantics may be used to associate a notion of robustness with ATs. Given a ranking semantics ρ, the degree assigned by ρ to the root argument within the tree can be interpreted as an indication of how stable its acceptance is under structural changes. Intuitively, higher values suggest that only substantial or well-targeted modifications of the underlying graph would affect the acceptability of the root, while lower values indicate that its status can be more easily undermined.

Given a candidate AT, its structural conditions from Definition 5 can be checked in polynomial time through standard graph operations. The extended attack relation can also be computed in polynomial time through graph reachability. Once this relation has been computed, checking whether the root can be labelled IN reduces to credulous acceptance under admissible semantics in a standard AF, which is NP-complete [17]. AT construction additionally requires selecting a suitable subset of arguments and relations. The ASP encoding guesses the arguments and relations included in the AT together with their acceptance status. Consequently, the number of candidate interpretations induced by the program can be exponential in the size of the input BAF. Computing a canonical AT further requires optimising the selected criterion over the valid solutions.

4. Acceptance Trees for Dialogue Systems

ATs provide a structured representation of the reasoning supporting a claim. Building on this representation, in this section we define a dialogue procedure that uses an AT to determine which arguments can be disclosed in response to the user's contributions. Every response selected by the procedure is therefore grounded in the underlying argumentative KB. Rather than generating arguments arbitrarily, the procedure only selects arguments already represented in the KB, ensuring that every reply remains logically justified and consistent with the underlying BAF. The following presentation focuses on the formal dialogue procedure and its response-selection policies, while the NLP components are described only to the extent required to explain how user utterances are mapped to arguments.

Each discussion topic is associated with an AT extracted from the corresponding BAF. The interaction starts from the root of the selected AT, representing the main claim under discussion. The conversation then unfolds incrementally, with each user contribution and each response selected by the procedure disclosing one argument at a time from the tree. In this way, each dialogue step progressively reveals a portion of the underlying justification, while preserving the structural and semantic properties established in the previous section.

The proposed dialogue procedure supports two interaction modes: PRO (decision support) and CON (persuasion). In PRO mode, the objective is to assist the user in decision making by reinforcing or clarifying a given position. In CON mode, the objective is to challenge the user's viewpoint and encourage a revision of it. As discussed later, the two modes rely on different response-selection policies while sharing the same underlying argumentative representation.

After each user utterance, the dialogue proceeds through two phases: argument recognition and response generation. Since reasoning is performed over the AT, a concrete implementation must first associate the user's natural-language input with one of its arguments. Each argument in the KB is assumed to be associated with one or more natural-language sentences expressing its content. One possible implementation strategy is to use a pre-trained Sentence-BERT model [18] to compute the semantic similarity between the user's message and these sentences, selecting the argument whose textual representation best matches the input. If no similarity score exceeds a predefined threshold, the utterance may be treated as outside the current KB and stored for possible future validation and integration.

Once the user's contribution has been mapped onto an argument of the AT, the response generation phase determines which undisclosed argument should be revealed next. The selection is performed according to a set of predefined rules that take into account the structure of the AT, the acceptability of arguments under the defence notion, the relation established by the user (attack or support), and the selected interaction mode. The objective is to produce replies that are logically valid, coherent with the ongoing discussion, and aligned with the intended conversational goal. The remainder of this section formalises the response-selection procedure.

The conversation can be seen as a sequence of controlled projections of the AT.[1] To incorporate the arguments corresponding to the user's input and the response selected by the procedure into the current projection of the AT, we apply a disclosure operation that introduces each new argument together with its associated attack or support relation.

Definition 6. *Let $B = \langle A, Att, Supp\rangle$ be a BAF and $B\mid_D$ its projection with respect to $D \subset A$, and consider an argument $c \in A \setminus D$. We define the disclosure of c in $B\mid_D$ as $B\mid_{D\cup\{c\}}$.*

The result of a disclosure operation is not, in general, guaranteed to be an AT for the root argument r, even when the original structure is itself an AT. As per Definition 6, a newly disclosed argument c may be added without any structural or semantic constraints. Consequently, c may be disconnected from the current projection or may affect the acceptability of already disclosed arguments, potentially leading to the rejection of the root. At the end of this section, we show that the proposed procedure avoids this issue by ensuring that structures obtained through the answer-selection rules remain ATs.

The response-selection procedure combines four key elements. First, the textual content of the user's contribution is mapped to a known argument b in the underlying AT. Second, the argument a to which the user is replying is determined. It may be explicitly selected by the user, assumed by default, for instance as the most recent argument disclosed by the procedure as in dialogue games [19], or inferred

[1] As usual, the projection of a BAF $B = \langle A, Att, Supp\rangle$ with respect to a subset of arguments $D \subseteq A$ is $B\mid_D = \langle D,\ Att \cap (D \times D),\ Supp \cap (D \times D)\rangle$.

Rule	a	b	Relation	Answer
P1	IN	OUT	$(b, a) \in Att_T$	$(c, b) \in Att_T$
P2	IN	IN	$(b, a) \in Att_T$	
P3	IN	OUT	$(b, a) \in Supp_T$	
P4	IN	IN	$(b, a) \in Supp_T$	$(c, b) \in Supp_T$ $(c, a) \in Supp_T$
P5	OUT	OUT	$(b, a) \in Att_T$	$(c, a) \in Att_T$
P6	OUT	IN	$(b, a) \in Att_T$	$(c, b) \in Supp_T$ $(c, a) \in Att_T$
P7	OUT	OUT	$(b, a) \in Supp_T$	$(c, b) \in Att_T$
P8	OUT	IN	$(b, a) \in Supp_T$	$(c, b) \in Supp_T$ $(c, a) \in Att_T$

Table 1
Answer selection rules in PRO mode.

automatically [20]. At the beginning of the interaction, the only available argument is the root of the AT, which is presented as the initial claim. Third, the relation between b and a is identified by determining whether b attacks or supports a. This information may be provided explicitly or inferred using learning-based methods [20]. Finally, the response strategy depends on the selected interaction mode. The PRO mode supports deliberation and decision making, whereas the CON mode challenges the user's stance and promotes persuasion. Separate rule sets are used for the two modes. To select a reply argument c, the procedure considers arguments in the AT that have not yet been disclosed. The selected argument must be acceptable under the adopted notion of defence to ensure that the resulting dialogue remains logically sound.

Definition 7. *Given an AT $T = \langle A_T, Att_T, Supp_T \rangle$, a set of disclosed arguments $D \subseteq A_T$, and the last user input $b \in D$ such that $(b, a) \in Att_T \cup Supp_T$ with $a \in D$, the response-selection procedure returns an argument $c \in A_T \setminus D$ satisfying $\mathcal{L}^T_{adm}(c) =$ IN, selected according to the rules in Tables 1 and 2 for the respective interaction modes.*

When a rule admits multiple responses, as in P4, P6, P8, and C7, the alternatives are interpreted disjunctively, and the procedure selects non-deterministically any undisclosed argument satisfying one of them. No preference order among the eligible responses is assumed. We first describe the rules for the PRO mode.

P1. The user provides an argument b that attacks an accepted argument a. This is interpreted as a disagreement with the current stance, and the procedure selects an argument c that challenges the user's position.

P2. Two accepted arguments attacking each other cannot occur under standard conflict-free semantics, hence this case is not considered.

P3. A rejected argument supporting an accepted one is impossible under the adopted notion of support.

P4. The user supports an accepted argument. This indicates agreement, and the procedure continues the discussion by selecting an additional supporting argument.

P5. The user uses a rejected argument to attack another rejected argument. The procedure selects an argument that reinforces the rejection of the target.

P6. The user provides an accepted argument attacking a rejected one. The procedure selects an argument that either supports the accepted argument or attacks the rejected one.

P7. The user supports a rejected argument with another rejected argument. The procedure blocks this line of reasoning by selecting an argument that attacks the supporting argument.

P8. The user supports a rejected argument with an accepted one. The procedure selects an argument that either strengthens the accepted argument or attacks the rejected one.

In CON mode, the procedure challenges the user's position when it conflicts with the root claim, which must remain accepted. Table 2 summarises the corresponding rules.

Rule	a	b	Relation	Answer
C1	IN	OUT	$(b, a) \in Att_T$	$(c, b) \in Att_T$
C2	IN	IN	$(b, a) \in Att_T$	
C3	IN	OUT	$(b, a) \in Supp_T$	
C4–C6				
C7	OUT	OUT	$(b, a) \in Supp_T$	$(c, b) \in Att_T$ $(c, a) \in Att_T$
C8	OUT	IN	$(b, a) \in Supp_T$	$(c, a) \in Att_T$

Table 2
Answer selection rules in CON mode.

C1. The user attacks an accepted argument with a rejected one. The procedure selects an argument that counterattacks the user's argument.

C2–C3. Cases corresponding to invalid configurations under the admissible semantics are excluded.

C4–C6. When the user aligns with the position represented by the accepted root by supporting accepted arguments or attacking rejected ones, the interaction is considered successfully concluded.

C7. The user supports a rejected argument with another rejected argument. The procedure selects an argument that attacks either component of the reasoning chain.

C8. The user supports a rejected argument with an accepted one. The procedure selects an argument that attacks the rejected argument.

We now establish key properties of the selection procedure. Although intermediate projections of an AT do not in general preserve its structure, the controlled disclosure strategy ensures that each step remains consistent.

Proposition 2. *Let $T = \langle A_T, Att_T, Supp_T \rangle$ be an AT rooted in r, let $T\mid_D$ be its projection, and let $c \in A_T \setminus D$ be the argument selected by the procedure in Definition 7. Then $T\mid_{D\cup\{c\}}$ is an AT rooted in r.*

Proof. The argument c is selected such that $\mathcal{L}^T_{adm}(c) = \text{IN}$. The rules ensure that any interaction involving c preserves the status of previously disclosed arguments, since updates only affect local relations within the tree. Structurally, c is introduced together with its unique connection to the already disclosed portion of the tree, preserving acyclicity and connectivity. Since each newly disclosed argument is linked to an existing node, reachability to the root is maintained. Therefore, all conditions of the AT definition continue to hold after disclosure, which concludes the proof. □

The dialogue is thus guaranteed to remain consistent throughout interaction, with each response preserving the underlying argumentative structure and maintaining a coherent justification of the root claim.

5. Related Work

Several tree-shaped structures have been proposed in the argumentation literature to explain or justify the acceptance of a claim. ATs are related to these approaches, but differ in their construction, their formal guarantees, and, most importantly, in their use as the underlying structure for incremental disclosure during a dialogue.

In Assumption-Based Argumentation, *abstract dispute trees* [21, 22] are used to establish the admissibility of a claim by alternating proponent and opponent moves. Similarly, *dialectical trees* in Defeasible Logic Programming [23] justify conclusions by recursively organising defeating arguments. Although both structures provide tree-shaped justifications, they are tightly coupled to their underlying argumentation formalisms. In contrast, ATs are defined over BAFs, where reasoning is driven by the combined presence of attack and support relations. This allows ATs to capture explanatory structures that cannot be naturally represented in abstract dispute trees or dialectical trees.

Closer to our setting, *bipolar dispute trees* [24] are also extracted from a BAF and explicitly account for both attack and support relations. However, their construction follows the classical dispute-tree paradigm: each node is assigned the role of either proponent or opponent, and the tree represents the recursive dialectical process through which a claim is defended against all possible attacks. ATs instead are rooted substructures of a BAF defined solely by structural constraints and the acceptance of a designated root argument, without introducing dialectical roles. This enables an incremental disclosure process that preserves the AT structure.

General dialogue trees [25] are built directly from the sequence of utterances exchanged in a dialogue, with each new utterance attached as a child of the previous one. The resulting structure is a direct trace of the interaction and is defined independently of any underlying argumentation formalism. As a consequence, it does not explicitly represent attack or support relations, and provides no guarantee that intermediate states preserve any notion of argumentative acceptability; such properties are only considered at the level of the complete dialogue.

A particularly close precedent to our dialogue procedure is provided by discussion games for argumentation semantics [26]. In these games, the acceptance of an argument under a given semantics is characterised through a structured exchange in which one participant defends the argument and the other challenges it. Similarly to our approach, arguments are disclosed incrementally through an alternating interaction. The two approaches, however, serve different purposes. Discussion games are proof procedures in which a winning strategy establishes the acceptance of the initial argument. In our setting, the acceptance of the root is established before the interaction through the construction of the AT, while the dialogue controls the progressive disclosure of a predefined substructure of a BAF. Moreover, PRO and CON denote alternative response-selection objectives within the procedure, rather than the roles of two players in a discussion game.

Our work also relates to research on argumentation-based approaches to persuasion and decision support, particularly those grounded in formal dialogue models. Prior studies have shown that computational argumentation provides effective tools for modelling persuasive interactions and structuring dialogues for exchanging and evaluating reasons [27, 28], as well as for supporting decision making through the comparison and justification of alternatives [29, 30]. Unlike these approaches, which often focus on abstract dialogue protocols or agent–agent interactions, our work considers a human-facing dialogue setting and grounds response selection in a structure extracted from a BAF. Moreover, existing approaches typically address either persuasion or decision support in isolation, whereas our framework supports both objectives within a unified model.

Several works have explored the use of NLP techniques to support human interaction in argumentation-based systems. For instance, [31] employ a Siamese BERT model to map user input to arguments based on semantic similarity, selecting the closest argument to maintain conversational flow. In contrast, our approach prioritises persuasion and decision support by selecting, among acceptable arguments, those that are most effective with respect to the dialogue goal.

In [32], a chatbot provides argument-based explanations for clinical treatment recommendations derived from patient records. Users may accept the recommendation or request further explanations, which are generated through predefined logic-based rules. Differently from this approach, we do not rely on task-specific rule sets but on a general argumentation structure to guide dialogue progression.

Other systems select responses based on users' concerns. The chatbot proposed in [33] exploits concerns to foster perspective change, selecting rebuttals associated with the most salient concern. A similar strategy is adopted in [34], where concerns correspond to values in a value-based AF and are identified through keyword matching. In contrast, our approach guides response selection solely on the arguments introduced during the dialogue and their relations within the BAF.

Argumentation-based chatbots have also been proposed in educational settings. For example, [35] presents a system that supports learning by analysing the structure of users' arguments and providing feedback. Unlike our approach, such systems do not select specific arguments to advance the dialogue, but rather present all acceptable arguments to the user.

Finally, the framework introduced in [36] combines LLMs with argumentation techniques to dynamically generate argumentation frameworks for claim verification. In contrast, our work assumes a predefined KB structured as a BAF and focuses on mapping user input to existing arguments, rather than inducing the argumentation structure itself.

6. Conclusions

We presented Acceptance Trees, tree-shaped structures extracted from BAFs that preserve the acceptance of a designated root argument and support the incremental disclosure of its justification. We also specified how ATs can underpin a dialogue procedure for decision support and persuasion. Within this procedure, natural-language inputs are mapped to arguments in the underlying KB, while responses are selected according to formal argumentation semantics and the chosen PRO or CON interaction mode. The present contribution focuses on the formal model, its executable ASP encoding, and the specification of the dialogue procedure.

Several extensions of the dialogue procedure are planned. First, we aim to support interactions in which the user can submit multiple arguments before receiving a reply. The response-selection procedure will then determine which argument to address first, for example by considering the most recent, the weakest, or the most strategically relevant one.

A second direction for future work concerns the computational and empirical evaluation of the approach. On the computational side, we will assess the scalability of the ASP encoding on BAFs of increasing size and density, considering both AT construction and the optimisation criteria used to obtain canonical ATs. On the empirical side, we will conduct user studies with real participants to assess the effectiveness of the dialogue procedure in both PRO and CON modes, focusing on perceived clarity, persuasiveness, and overall interaction quality.

Generating the ATs that underpin the dialogue procedure requires selecting a substructure in which the designated root argument is accepted. This selection can be viewed as a structural modification of the original BAF and is therefore related to enforcement approaches developed for classical AFs [13]. We plan to investigate this connection systematically by studying enforcement techniques for BAFs under different argumentation semantics.

In addition, we plan to extend the framework to quantitative argumentation models, such as weighted or probabilistic BAFs, to support richer KBs and more fine-grained reasoning. We also aim to extend the dialogue procedure to multi-user interactions, allowing several participants to engage in collaborative or competitive debates. Finally, a future implementation should be able to process user arguments that are not yet represented in the KB and support their subsequent validation and integration.

A further direction for future work concerns the semantic comparison of ATs. While different ATs may justify the same root argument, it is natural to ask whether they can be ordered according to the strength of the justification they provide by exploiting graded or ranking-based semantics for bipolar argumentation. This would allow the notion of robustness of an AT to be grounded semantically rather than purely structurally. An interesting open question is whether such an ordering is consistent across different ranking semantics, or more realistically, across classes of semantics satisfying common rationality principles [37].

Acknowledgments

The authors are members of the INdAM Research group GNCS and of Consorzio CINI. This work has been partially supported by: INdAM - GNCS Project, CUP E53C25002010001; University of Perugia - Fondo Ricerca di Ateneo (2020, 2022) – Projects RATIONALISTS, “Civil Safety and Security for Society”.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.6 in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

[1] P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre, Handbook of Formal Argumentation, College Publications, 2018.

[2] K. Cyras, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative XAI: A survey, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 4392–4399. URL: https://doi.org/10.24963/ijcai.2021/600. doi:10.24963/IJCAI.2021/600.

[3] F. Castagna, N. Kökciyan, I. Sassoon, S. Parsons, E. Sklar, Computational argumentation-based chatbots: A survey, J. Artif. Intell. Res. 80 (2024) 1271–1310. URL: https://doi.org/10.1613/jair.1.15407. doi:10.1613/JAIR.1.15407.

[4] M. Thimm, S. Villata, The first international competition on computational models of argumentation: Results and analysis, Artif. Intell. 252 (2017) 267–294. URL: https://doi.org/10.1016/j.artint.2017.08.006. doi:10.1016/J.ARTINT.2017.08.006.

[5] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the second international competition on computational models of argumentation, Artif. Intell. 279 (2020).

[6] S. Bistarelli, L. Kotthoff, J.-M. Lagniez, E. Lonca, J.-G. Mailly, J. Rossit, F. Santini, C. Taticchi, The third and fourth international competitions on computational models of argumentation: Design, results and analysis, Argument Comput. Pre-press (2024) 1–73.

[7] M. Järvisalo, T. Lehtonen, A. Niskanen, ICCMA 2023: 5th international competition on computational models of argumentation, Artif. Intell. 342 (2025) 104311. URL: https://doi.org/10.1016/j.artint.2025.104311. doi:10.1016/J.ARTINT.2025.104311.

[8] L. Amgoud, C. Cayrol, M. Lagasquie-Schiex, P. Livet, On bipolarity in argumentation frameworks, Int. J. Intell. Syst. 23 (2008) 1062–1093.

[9] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–358. URL: https://doi.org/10.1016/0004-3702(94)00041-X. doi:10.1016/0004-3702(94)00041-X.

[10] C. Cayrol, M. Lagasquie-Schiex, Bipolarity in argumentation graphs: Towards a better understanding, Int. J. Approx. Reason. 54 (2013) 876–899. URL: https://doi.org/10.1016/j.ijar.2013.03.001. doi:10.1016/J.IJAR.2013.03.001.

[11] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Answer Set Solving in Practice, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012. URL: https://doi.org/10.2200/S00457ED1V01Y201211AIM019. doi:10.2200/S00457ED1V01Y201211AIM019.

[12] S. A. Gaggl, Solving Argumentation Frameworks Using Answer Set Programming, Technical Report, DBAI, TU Wien, 2009. URL: https://www.dbai.tuwien.ac.at/proj/argumentation/papers/Gaggl_Thesis.pdf.

[13] R. Baumann, S. Doutre, J. Mailly, J. P. Wallner, Enforcement in formal argumentation, FLAP 8 (2021) 1623–1678. URL: https://collegepublications.co.uk/ifcolog/?00048.

[14] C. Cayrol, M.-C. Lagasquie-Schiex, Gradual valuation for bipolar argumentation frameworks, in: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer, 2005, pp. 366–377.

[15] A. Pazienza, S. Ferilli, F. Esposito, On the gradual acceptability of arguments in bipolar weighted argumentation frameworks with degrees of trust, in: International Symposium on Methodologies for Intelligent Systems, Springer, 2017, pp. 195–204.

[16] C. Fasciano, G. Loseto, A. Pinto, M. Ruta, F. Scioscia, A propagation-based ranking semantics in explainable bipolar weighted argumentation, Engineering Applications of Artificial Intelligence 142 (2025) 109767.

[17] W. Dvorák, P. E. Dunne, Computational problems in formal argumentation and their complexity, FLAP 4 (2017). URL: http://www.collegepublications.co.uk/downloads/ifcolog00017.pdf.

[18] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990.

[19] C. Reed, Dialogue frames in agent communication, in: ICMAS, IEEE Computer Society, 1998, pp.

246–253.
[20] M. Lippi, P. Torroni, Argumentation mining: State of the art and emerging trends, ACM Trans. Internet Techn. 16 (2016) 10:1–10:25. URL: https://doi.org/10.1145/2850417. doi:10.1145/2850417.
[21] P. M. Dung, R. A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, Artificial Intelligence 170 (2006) 114–159.
[22] X. Fan, F. Toni, On computing explanations in argumentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015.
[23] A. J. García, G. R. Simari, logic programming: an argumentative approach, Theory and practice of logic programming 4 (2004) 95–138.
[24] Z. Zeng, C. Miao, C. Leung, Z. Shen, J. J. Chin, Computing argumentative explanations in bipolar argumentation frameworks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press, 2019, pp. 10079–10080. URL: https://doi.org/10.1609/aaai.v33i01.330110079. doi:10.1609/AAAI.V33I01.330110079.
[25] L. Ho, S. Schlobach, A dialogue framework for logic-based argumentation., in: ArgXAI@ COMMA, 2024, pp. 41–55.
[26] M. Caminada, Argumentation semantics as formal discussion, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, volume 1, College Publications, 2018, pp. 487–518.
[27] A. Hunter, Towards a framework for computational persuasion with applications in behaviour change, Argument Comput. 9 (2018) 15–40. URL: https://doi.org/10.3233/AAC-170032. doi:10.3233/AAC-170032.
[28] X. Fan, F. Toni, Mechanism design for argumentation-based persuasion, in: B. Verheij, S. Szeider, S. Woltran (Eds.), Computational Models of Argument - Proceedings of COMMA 2012, Vienna, Austria, September 10-12, 2012, volume 245 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2012, pp. 322–333. URL: https://doi.org/10.3233/978-1-61499-111-3-322. doi:10.3233/978-1-61499-111-3-322.
[29] J. Müller, A. Hunter, An argumentation-based approach for decision making, in: IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012, IEEE Computer Society, 2012, pp. 564–571. URL: https://doi.org/10.1109/ICTAI.2012.82. doi:10.1109/ICTAI.2012.82.
[30] L. Amgoud, Argumentation for decision making, in: G. R. Simari, I. Rahwan (Eds.), Argumentation in Artificial Intelligence, Springer, 2009, pp. 301–320. URL: https://doi.org/10.1007/978-0-387-98197-0_15. doi:10.1007/978-0-387-98197-0_15.
[31] C. Hauptmann, A. Krenzer, J. Völkel, F. Puppe, Argumentation effect of a chatbot for ethical discussions about autonomous AI scenarios, Knowl. Inf. Syst. 66 (2024) 3607–3637.
[32] F. Castagna, A. E. Garton, P. McBurney, S. Parsons, I. Sassoon, E. I. Sklar, Eqrbot: A chatbot delivering EQR argument-based explanations, Frontiers Artif. Intell. 6 (2023).
[33] E. Hadoux, A. Hunter, Comfort or safety? gathering and using the concerns of a participant for better persuasion, Argument Comput. 10 (2019) 113–147.
[34] L. A. Chalaguine, A. Hunter, A persuasive chatbot using a crowd-sourced argument graph and concerns, in: COMMA, volume 326 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 9–20.
[35] T. Wambsganss, S. Guggisberg, M. Söllner, Arguebot: A conversational agent for adaptive argumentation feedback, in: Wirtschaftsinformatik, AISeL, 2021.
[36] G. Freedman, A. Dejl, D. Gorur, X. Yin, A. Rago, F. Toni, Argumentative large language models for explainable and contestable claim verification, in: T. Walsh, J. Shah, Z. Kolter (Eds.), AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, AAAI Press, 2025, pp. 14930–14939. URL: https://doi.org/10.1609/aaai.v39i14.33637. doi:10.1609/AAAI.V39I14.33637.
[37] L. Amgoud, J. Ben-Naim, Ranking-based semantics for argumentation frameworks, in: Interna-

tional Conference on Scalable Uncertainty Management, Springer, 2013, pp. 134–147.