Applications of Argumentation-based dialogues

Andreas Xydis
Lincoln Institute for Agri-Food Technology, University of Lincoln, United Kingdom
AXydis@lincoln.ac.uk

Federico Castagna
Department of Computer Science, Brunel University, U.K.
Federico.Castagna@brunel.ac.uk

Elizabeth I Sklar
Lincoln Institute for Agri-Food Technology, University of Lincoln, United Kingdom
ESklar@lincoln.ac.uk

Simon Parsons
Lincoln Centre for Autonomous Systems, University of Lincoln, United Kingdom
SParsons@lincoln.ac.uk

Abstract

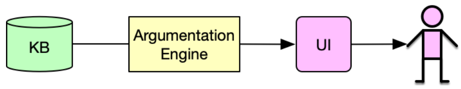
This chapter reviews work on applications of argumentation-based dialogue. It takes both a broad view of dialogue and a broad view of what constitutes an application. It considers the full range of software tools that would be needed in constructing a software system that is capable of engaging in argumentation-based dialogue, along with complete applications, and includes both work that builds on formal models of dialogue, and that is more inspired by recent work on chatbots from natural language processing.

1 Introduction

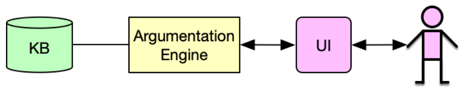
Arguments are intrinsically intertwined with the notion of dialogue, since dialogue can be characterized as the interplay of arguments. As argued in an earlier Handbook chapter: *“ [...] it is the idea of dialogue as an exchange between two or more individuals, an exchange which captures features of what would be informally called an “argument”. That is, dialogue as the exchange of reasons* [*i.e., arguments*] *for or against some matter.”*[30]. Notice that, due to their potential expressivity (despite being subject to specific restrictions), dialogues have been advocated and often chosen as the standard communication protocol within the multi-agent system paradigm that views computation as predominately led by interaction [128]. Indeed, this new paradigm required the design of an appropriate means of communication between such intelligent agents [131], thus acknowledging the importance that dialogue exhibits in any kind of interplay, whether it occurs among humans, computational entities or both.

In this chapter we build on [30], which covered some of the theoretical basis of argumentation-based dialogues by discussing applications of argumentation-based dialogues. We review and analyse a broad spectrum of proposed and existing implementations, ranging from fully-fledged software suites to rough sketches of architectures at an early stage of deployment. The key element in the distinction is that all this work is focused on the deployment of argumentation-based dialogue, rather than the development of new argumentation models.

Figure 1 illustrates our definition of argumentation-based dialogue systems and clarifies the scope of this chapter. We consider systems that have some kind of underlying *Knowledge Base (KB)*, some kind of *Argumentation Engine*, which builds arguments from the contents of the KB and/or computes the extensions of a set of arguments according to some semantics, and some kind of *User Interface (UI)*. A number of implemented argumentation systems have provided a UI whereby the user can interrogate the underlying argumentation engine and KB, but this functionality serves only to help the user gain understanding of the reasoning performed by the argumentation engine and the information it used for reasoning. While the user may be able to query the engine and KB, the user does not have the ability to *change* anything in the



(a) One-way system: Interactive interface



(b) Two-way system: Dialogue

Figure 1: Scope of systems discussed here: "one-way" systems provide a user interface for interrogating the underlying argumentation engine and knowledge base (KB), but do not allow the user to change the knowledge base in the reasoning system; "two-way" systems support true *dialogue*, in the sense that the user is considered part of the system and can influence the knowledge and rules employed by the system. The direction of the arrows in the figure denote flow of information.

KB. This type of *one-way* system is illustrated in Figure 1a. The second type of system—the one which we focus on in this chapter—is specifically a *two-way* system, whereby the user engages in a *dialogue* with the system and therefore has the ability to *change* information in the system's KB and hence affect the arguments that the system can construct[1]. In some cases, the user also has the ability to change the behaviour of the argumentation system, by altering the rules used by the reasoner, through dialogue. This type of *two-way* system is illustrated in Figure 1b. Here, in this chapter, we mainly focus on the second type of system (but include a couple of examples of one way systems).

Another set of distinctions can be drawn between the groups of indi-

[1]A number of argumentation-based dialogue systems make use of the notion of a "commitment store", which, in the context of Figure 1, would hold information presented by the user. Depending on the system, this commitment store might, from a theoretical perspective, be considered to be distinct from the knowledge base of the system. However, the contents of the commitment store can often be used by the argumentation system in the construction of arguments, and in such cases we would consider it to be a subset of the KB in a two-way system.

viduals that are engaged in the dialogue. A participant can be a human or an agent — we make no distinction between agents that are purely software, and agents that have a physical embodiment, such as a robot. As a result, we can imagine three kinds of dialogue: those that involve only humans, those that involve only agents, and those human-agent dialogues that involve both kinds of participant. We also consider that dialogues may only involve a single participant. For example, many of us are familiar with the kind of internal conversation that provides a mechanism for reflecting on some position or for ensuring that an argument will be convincing to an audience. We call such dialogues "human-self". Similarly we recognise "agent-self" dialogues, and point out that such dialogues are one way of computing argument acceptability [41, 43, 48].

2 Theoretical Foundations

In this section we provide an overview of the fundamental notions underlying the remainder of the chapter. In particular, at first, we discuss argumentation frameworks and then formal models of dialogues. This then leads into Section 3 which sketches some of the components, such as argumentation solvers, that are used to construct applications of argumentation-based dialogues.

2.1 Argumentation frameworks

Dung's abstract argumentation theory [71] has a good claim to be the most known and used formalism in computational argumentation for AI at this point. The basic formalism from [71] has been extended in a number of different ways (e.g., see [3, 4, 14]). Dung's paper [71] can be recognised as a supporting and analytical tool for non-monotonic reasoning [15] due to the fact that arguments are represented as abstract entities, providing also important insights regarding the semantic acceptance of arguments. Specifically, Dung introduced a framework (known as an *abstract argument framework* — AAF), which is used to depict the attacking relationships (represented as directed edges) between arguments (nodes) in a graph G. A set of arguments S appearing in G is called *conflict-free* if and only if there are no arguments $A, B \in S$ such that A attacks B or B attacks A. Based on G more sets of arguments are defined with specific characteristics (called *extensions*) that determine which arguments are *acceptable* according to different semantics such as *complete, preferred* and *grounded* (see [71] for details). Alternatives for computing extensions in AAFs have also been developed (e.g., the *labelling function* described in [40]).

Although the analysis of arguments at the abstract level can provide many insights into the way that rational[2] arguers can and should behave, abstract argumentation is not always enough when tackling real argumentation problems. For example, using abstract argumentation we cannot examine how arguments are instantiated, how conclusions are inferred, what is the nature of the attacks between arguments or how arguments (and their supports or attacks) change over time. To support the justification process more naturally and examine an argument structurally, we need to access its internal parts. Thus, structured approaches to argumentation seem more appropriate to exploit.

Different frameworks for structured argumentation have been established, such as ABA [32, 72], $ASPIC^{+}$ framework [138], $DeLP$ [88] and *deductive argumentation* [17], for instantiating arguments with some internal structure (see [13] for details). These arguments are typically instantiated on the basis of a *knowledge base* (KB) — that contains certain and/or defeasible (i.e., uncertain) information (often called *premises*) represented in a logical language — and the application of *inference rules* to premises, leading to the *conclusion* of an argument — represented in the same logical language. Note, an argument may consist of multiple sub-arguments in which case *intermediate conclusions* are also part of an argument. Additionally, two types of inference rules may be defined: (1) *strict rules* whose inferences are certain to hold; and (2) *defeasible rules* whose inferences are, presumably, a consequence of the premises. Inference rules are explicit when more than one type are used, but may be implicit when only one type is employed. Templates that represent structures of common kinds of presumptive arguments used in everyday discourse as well as special contexts (e.g., legal argumentation), called *argument schemes*, may be perceived as a special type of inference rules since they connect the premises and the conclusion of an argument. The internal structure of arguments allows for different

[2]For a range of different instantiations of the concept "rational".

kinds of attacks: (1) attacking the premises of an argument (usually called *undermine*); (2) attacking the application of defeasible rules used to infer a conclusion (usually called *undercut*); and (3) attacking the conclusion–if this is inferred using a defeasible rule — of the argument (usually called *rebuttal*).

In real-life, however, human agents often use 'incomplete' arguments known as *enthymemes*. In the argumentation literature there have been works on frameworks which explore how an enthymeme can be constructed from the intended argument and how the intended argument can be reconstructed from an enthymeme. For example, in [102] the author uses common knowledge (CK) between agents to show how real arguments can be encoded by the sender and decoded by a recipient. Specifically, an agent may omit premises of their intended argument that they assume to be part of CK between the agents. The recipient aims to understand these missing premises by referring to CK. In [29] (an extension of the work presented in [102]), the authors propose a formal framework for constructing and reconstructing enthymemes based on relevance theory [177] which is grounded in two principles: *maximising cognitive effect* and *minimising cognitive effort*. There, two classes of enthymemes are defined: (1) the *implicit support enthymemes* which are enthymemes that do not include all the premises needed to entail the claim of the argument they are constructed from; and (2) the *implicit claim enthymemes* which are enthymemes missing some of the premises of the argument they are constructed from as well as the claim of that argument. In Section 6.1, we examine more works on enthymemes, which concentrate particularly on the handling of enthymemes in dialogues.

2.2 Formal models of dialogues

Both abstract and structured approaches to argumentation define binary attack (or defeat) relations between arguments where the claims of the winning (acceptable) arguments in the argument framework AF identify the non-monotonic inferences from the belief base instantiated in AF. These approaches, initially defined for single agent (monological) reasoning, can be generalised to dialogical models of distributed non-monotonic reasoning in which two or more agents exchange arguments and other locutions.

Dialogue types	**Initial situation**	**Individual goal**	**Dialogue goal**
Persuasion	Conflict of opinions	Persuade other party	Resolve or clarify issue
Inquiry	Need for proof	Find and verify evidence	Prove/disprove hypothesis
Information-seeking	Need Information	Acquire or give information	Exchange information
Negotiation	Conflict of interests	Get what you most want	Reasonable settlement both can live with
Deliberation	Dilemma or practical choice	Co-ordinate goals and actions	Decide best available course of action

Table 1: Types of dialogues proposed in [192].

Walton and Krabbe's work [192] was one of the most influential regarding the typology of primary dialogue types. Each type of dialogue depends on the initial information that participants have, their individual goals and the objective of the dialogue (see Table 1). A number of these types of dialogue have been studied in detail by the argumentation community. For example, see [158] for persuasion, [79] for information-seeking, [28] for inquiry, [162] for negotiation and [130] for deliberation. Walton and Krabbe's list is not intended to be exhaustive, and not only is it possible to identify kinds of dialogue beyond those in Table 1[3], but also new dialogues can be formed by combining types of dialogues from Table 1.

Here, we briefly describe common constituents of such dialogue systems.

[3]For example [62] does this by considering different combinations of initial situation.

2.2.1 Constituents of dialogues

As described in [30], the most common constituents of argumentation-based dialogues found in the literature are *moves, dialogue history, protocol* and, possibly, *strategies*[4]. We review each of these here.

Moves. Although different variations of *moves* can be found in the literature, there are three basic pieces which constitute a move. These are the *sender* of the move[5], the *locution* of the move (which indicates the *type* of the move, i.e. what an agent is allowed to utter using this move), and the *content* of the move. Different locutions may be used in different dialogue systems, but a common set include the following: *assert*, *accept*, *challenge*, *question*, *since* and *retract*. They allow for claims and their supporting arguments to be stated (or retracted), arguments to be requested and questions to be asked. The content of a move can be a formula ϕ built from some logical language, or even *null*.

Dialogue history. The *dialogue history* represents the moves made during the dialogue and it is usually formalised as a non-empty sequence of these moves. In most cases moves are indexed by the step of the dialogue, where d_k denotes the length k of the dialogue.

Protocol. The *protocol* of a dialogue determines the legal moves an agent can make during a dialogue. Although it is impractical to designate the moves permitted for every possible dialogue state, simpler rules can indicate what kind of move an agent is allowed to make. Such rules can be turn-taking rules, relationship between locutions (e.g., if an agent utters a *question* move, their interlocutor can reply only with a *since* move) or commitment rules stating conditions upon which moves can be made (e.g., if an agent has asserted ϕ and has not retracted it, then

[4] Note that while we recognise that modelling the beliefs of other agents [170], and belief revision as a result of dialogue [153] are both important with relation to dialogues, we consider them to be out of scope for this chapter, not least because belief revision and argumentation are studied at length elsewhere [78].

[5] The sender is often considered to be either the *proponent* of an argument or the opponent, since most work on dialogue considers just two participants (assuming easy generalisation to many) and a more or less adversarial stance where one agent (proponent) is trying to have the other accept the argument that they are making.

asserting $\neg\phi$ will produce an inconsistent *commitment store* CS for the agent).

Strategy. A *strategy* is a mechanism for deciding an agent's move. This can be determined by the agent's objective (e.g., preserving rationality principles or "winning" the dialogue). Formally, a strategy S_{Ag} of an agent Ag may be perceived as a function $S_{Ag} : D \times K_{Ag} \longrightarrow 2^M$, where D denotes the dialogue history, K_{Ag} the private knowledge base KB of Ag and M the set of moves. If there is no probability distribution over the possible dialogue moves of agents then a strategy is *deterministic*, whereas a strategy which returns only a single move is called *decisive*. A common practice for dialogue systems that employ strategies is for these to depend on the previous move made in the dialogue.

2.2.2 Persuasion dialogue systems

In persuasion dialogues, two or more participants try to resolve a conflict of opinion, each trying to persuade the other participant(s) to adopt their point of view. Many papers examine persuasion dialogues; and different features in combination with such dialogues are investigated, such as opponent modelling (e.g., [92, 147]), planning (e.g., [26]), decision trees for strategising (e.g., [93]), probabilities (e.g., [105]) and natural language processing (e.g., [54, 163]). We briefly introduce some important formal models of dialogue systems [158, 159] that capture persuasion so that it is easier for the reader to understand the main aspects of practical applications concerning persuasion dialogues presented later.

Walton and Krabbe's paper [192] describes the "Permissive Persuasion Dialogues" (PPD) system (amongst other systems capturing different types of dialogues). In PPD, dialogues have no context, and include two participants (P and Op) which may declare assertions and concessions in an implicit preparation phase before the dialogue commences. Each one is considered the proponent of their own assertions and opponent of the other participant's initial assertions. The communication language includes challenges, (tree-structured) arguments, concessions, questions, resolution demands, and two types of retraction locutions for commitments. The logical language used is propositional logic, and the inference rules are deliberately incomplete to reflect the complexity of natural language. Participants commit to the premises of the arguments they move, but arguments may be incomplete, leaving room for further exploration (these may also be perceived as enthymemes, but a further discussion on dialogue systems that deal with enthymemes will follow later).

The protocol is guided by the participants' CSs and the content of the move made in their last turn. P starts the dialogue, and in the first turn, both P and Op either concede or challenge each other's initial assertions. Each turn, a participant is obliged to reply to all moves made in the other player's last turn, except concessions and retractions. Multiple replies are allowed, and alternative arguments for the same assertion can be made. Counterarguments are not permitted. The protocol is non-deterministic, multi-move and multi-reply, but postponement of replies is not allowed. Challenges, concessions, retractions and questions are always related to commitments. A participant cannot challenge or concede their own commitments. Inconsistent commitments can be resolved, and implications between commitments may require concessions or retractions. The outcome of the dialogue is determined by the participants' commitments, and the dialogue terminates after a predetermined number of turns.

In [155], a dialogue system between two "players" is described. The dialogue concerns a single topic and each participant has their own KB which itself may be inconsistent. The communication language allows for participants to move claims, challenges, and concessions during the dialogue, but there is no explicit reply structure. Claims can pertain to individual propositions or sets of propositions, and the logic employed is non-monotonic. Arguments are classical proofs from consistent premises, and arguments attack other arguments by negating a premise of their target. Conflict between arguments is resolved using a preference relation on the premises. The system utilizes grounded semantics to decide acceptability of arguments. Arguments can be implicitly moved as a claim ψ replying to a challenge of another claim ϕ, given that ψ is consistent and ψ contradicts ϕ. The system defines commitment rules, but (contrary to Walton and Krabbe's system [192] discussed above) these rules neither determine legal moves nor the outcome of the dialogue. A CS is only used as a supplementary KB to access an interlocutor's knowledge revealed during the dialogue.

An important aspect introduced in this system is the *assertion* and *acceptance* attitudes of players, which they must adhere to throughout the dialogue. These attitudes are defined in relation to the player's private KB, which does not change during the dialogue, and both players' CSs, which may change during the dialogue (see [155] for details). Players' attitudes influence their moves during the dialogues. Although there are works (e.g., [158, 192]) defending the idea that a dialogue protocol should only enforce coherence of dialogues, [155] argues that a dialogue protocol should refer to private KBs to ensure rationality and honesty of players. As a result, a formal definition of a protocol is given where the assertion and acceptance attitudes of players partly dictate the legal moves of the system, as well as termination of a dialogue. The winner of a dialogue is not defined, but the possible outcomes are defined in terms of the propositions claimed and conceded by participants. Finally, the protocol is unique-move, unique-reply and deterministic, with some exceptions (see [155] for details).

In [157], Prakken establishes a general dialogue framework, assuming two participants, whose purpose is to formally describe the components needed to formalise any kind of dialogue. Specifically, his initial dialogue framework is general enough to capture various kinds of dialogue from Walton and Krabbe's typology [192], whereas later he specifies locutions and rules which are used to model persuasion dialogues (called *liberal* dialogues). Prakken is non-committal on certain specifics, except for an explicit reply structure between moves. Moreover, he explores different protocols (of varying degrees of complexity) for regulating dialogues and the belief bases of participants do not influence the dialogues' protocol.

Prakken also defines the *dialogical status* of moves made in a dialogue so that: (a) different turn-taking and termination rules are examined as well as the relevance of moves; and (b) a correspondence is established between the dialogical status of the initial move of a dialogue (whose content is the topic of the dialogue) and the justified arguments in supporting the dialogue topic. An argument is presented as a tree whose nodes are elements of the logical language, the edges between them depict either strict or defeasible inference rules, the root of the tree is the conclusion of the argument and its leaves are the argument's premises, similar to $ASPIC^+$. The locutions introduced for modelling persuasion dialogues in [157] are $claim, why, concede, retract$ and $argue$. The dialogical status of a move can be either IN or OUT. The author represents a dialogue as a tree where each node n is a move, and a child of n (if any) is a reply to n.

Additionally, the notion of *logical completion* of a dialogue is introduced, which means that if there is an argument A that can be constructed from the content of the locutions exchanged during the dialogue which defeats an argument B asserted by a participant, then A is moved against B for each occurrence of B in the dialogue (since an argument may be moved multiple times in a dialogue; and each time is represented as a different instance in the dialogue tree). Furthermore, the participant Ag making the first move m_0 in the dialogue is the *winner* of the dialogue if and only if m_0 is IN, otherwise its counterpart Ag' is the winner of the dialogue. Termination is defined as the situation where a player is supposed to move but has no legal moves. Alternative and postponed replies are allowed, and the instantiations of protocols described are multi-move and multi-reply. Finally, it is worth mentioning that the author produces some soundness and fairness results stating that, for a finite logically completed liberal dialogue, Ag wins the dialogue (under the grounded semantics) if and only if using the contents of the locutions exchanged during the dialogue, a justified argument in support of the topic of the dialogue (i.e., the content of the first move, m_0, made in the dialogue) can be constructed.

2.2.3 Inquiry dialogue systems

In inquiry dialogues, participants collaborate to answer some question(s) that they could not answer on their own. Comparatively little work has been done on inquiry dialogue protocols that employ argumentation. We briefly describe two important papers related to this field. Later on in this chapter, we present practical applications concerning inquiry dialogues that implement characteristics discussed below.

As mentioned earlier, [155] presents some protocols for different types of dialogues. Most of the characteristics discussed in this chapter, in Section 2.2.2, for the same paper [155] also hold for inquiry dialogues, i.e. the number of participants and locutions allowed; the logic is non-monotonic; arguments are still classical proofs from consistent premises and not directly moved, but implicitly; and the commitment rules do not determine legal moves nor the outcome of the dialogue. However, since agents work together, they do not attack each other's arguments but only challenge them. Agents' assertion and acceptance attitudes influence the protocol of the dialogue in this case, too. Additionally, two different protocols are described, since the first one (assert) is so simple that it includes flaws which the authors try to address (e.g., a proof might not be allowed to be found even though it is available to the agents if they moved different sets of assertions). Although the second protocol (accept) presented deals with the issues of the first protocol, again the authors mention that there is room for improvement (e.g., in the second protocol, only one agent dictates assertions, which are also restricted to be connected to what is already uttered). Finally, although the first protocol presented is unique-move and unique-reply, the second one allows for multiple moves and replies.

In [28], a general inquiry dialogue system between two participants is introduced where a strategy for each agent for picking a unique move to make in each step of the dialogue is also developed. Specifically, at first, the authors give a general definition of a dialogue which allows for other types to be considered within their framework, whereas later they provide protocols for two different types of inquiry dialogues: (1) an *argument inquiry dialogue*, which allows participants to share knowledge to jointly construct arguments; and (2) a *warrant inquiry dialogue*, which allows participants to share knowledge to jointly construct dialectical trees (i.e., a tree with an argument at each node in which a child node is a counterargument to its parent). In an argument inquiry dialogue, the agents exchange beliefs in order to jointly construct arguments for a particular claim, but the acceptability of the arguments constructed cannot be determined. However, in a warrant inquiry dialogue, the acceptability of a particular argument is examined by jointly constructing a dialectical tree that collects all the arguments that may be relevant to the acceptability of the argument in question.

The authors use $DeLP$ to represent not only beliefs and arguments, but also preferences over arguments to decide successful attacks. The dialectical status of an argument in a dialectical tree is either D, for defeated arguments, or U, for undefeated ones. The locutions allowed are: *open*, *assert* and *close*. For an argument inquiry dialogue, the topic of the dialogue is a defeasible rule, whereas for a warrant inquiry dialogue, its topic is a defeasible fact. The termination of a dialogue (in both cases) is defined as the consecutive appearance of two close moves. Essentially, this means that both participants must agree to the termination of the dialogue (as they alternate moves). Both agents have a CS, but there is also a shared *query store*, defined as the set of literals that could help construct an argument for the consequent of the topic of an argument inquiry dialogue (during the dialogue, participants try to provide arguments for the literals in the query store). The outcome of an argument inquiry dialogue is defined as the set of all arguments that can be constructed from the union of the CSs and whose claims are in the query store, whereas the outcome of a warrant inquiry dialogue is determined by the dialectical tree that is constructed from the union of the CSs: the topic of the dialogue is warranted if and only if the root of the dialectical tree is undefeated.

2.2.4 Information-seeking dialogue systems

In information-seeking dialogues, the goal of the dialogue is the exchange of information where a participant wants to acquire some information they are not aware of from their interlocutor who tries to fulfill that request. To the best of our knowledge there are not many works on dialogue systems designed specifically for information-seeking dialogues. Instead, general dialogue systems have been examined as to how they could be used to instantiate this type of dialogues. Below we present how an information-seeking framework has been considered in [155].

Most of the characteristics discussed in this chapter, in Sections 2.2.2 and 2.2.3, for the paper [155] also hold for information-seeking dialogues, i.e. the number of participants and locutions allowed; the logic is non-monotonic; arguments are still classical proofs from consistent premises and not directly moved, but implicitly; and the commitment rules do not determine legal moves nor the outcome of the dialogue. However, in this case, the dialogue starts with a question from participant A towards their interlocutor B regarding a proposition p. The dialogue is similar to an inquiry dialogue, where agents cannot attack each other's arguments, but only challenge them and provide support for them. Agents' assertion and acceptance attitudes influence the protocol of the dialogue in this case, too. Later on, authors discuss interesting properties that characterise their protocol, some of which are true for any assertion and acceptance attitudes of participants, whereas some others depend on these attitudes.

2.2.5 Negotiation dialogue systems

In negotiation dialogues, participants try to resolve a conflict of interest by reaching a deal that all the parties can live with. Although much work has been done in the development of theoretical negotiation dialogue frameworks, only a little work has been done on practical applications that consider negotiation dialogues which account for human agents. We briefly present two influential works in this topic, and later in Section 4.3.5 we examine related applications found in the literature.

In [154] an argumentation-based framework for negotiation dialogues is proposed where the associated protocol that governs the dialogue is presented as a state machine. The locutions of this protocol are: *proposal* (used to open the dialogue suggesting a solution to the problem that the agents face, or offer a proposal at a different stage in the dialogue), *critique* (used to provoke an alternative proposal), $counter-proposal$ (which is a proposal that is made as a response to a previous one), *accept* (used to accept a proposal showing that an agreement is reached and the dialogue terminates) and *withdraw* (used by a participant to leave the dialogue showing that no agreement is reached and the dialogue terminates). Note that agents may make counter-proposals without waiting for a response to a previous one, and the participants of the dialogue are assumed to be two although they can be more.

While describing their introduced locutions and protocol, the authors refer to the notion of *explanation* which they define as additional information explaining why a proposal, counter-proposal or critique was made. Essentially they present a pair $p = (\Gamma, \phi)$ consisting of an utterance ϕ and an explanation Γ as an argument where Γ is a set of formulae available to an agent. To construct arguments, the authors use classical first-order logic. Arguments are used as part of the content to the locutions described above. They also define rebut and undercut from an argument A towards an argument B as an attack from A to the conclusion and premise(s) of B, respectively, but also mention that attacks on stated inference rules can also take place without extending their discussion on this topic.

Notice that a reason for using first-order logic, is because the agents' architecture follows the BDI model (Belief-Desire-Intention), thus each of these components can be represented as a predicate in the communication language of the agents. Additionally, because of their BDI model, they define conflicts between agents as agents having opposite intentions, or an agent intending to change the mental state of their interlocutor. Finally, classes of acceptable arguments are also defined, so that the agent can determine how strongly it objects to a proposal as well as evaluate it internally before sending it as a proposal to the other agent.

In [5], the authors provide another protocol for negotiation dialogues, which is based on abstract argumentation to instantiate arguments. Specifically, an argument is defined as a pair $A = (H, h)$ where h is a formula of a propositional language L, and H is a set of formulae of L such that H is consistent, $H \vdash h$ and H is minimal with respect to set inclusion. The KBs of the participants may be inconsistent and an *undercut* between arguments is defined as a case where the conclusion of one argument A contradicts one of the elements of the support of another argument B. A preference ordering between arguments is also taken into account to determine successful attacks and, thus, acceptable arguments.

Later, a protocol is given which assumes two participants, describes the legal moves of the dialogue and defines an argument dialogue as a sequence of such moves. Argument dialogue trees are also defined, where each branch of this finite tree is an argument dialogue, and winning criteria are also discussed. Additionally, the authors assume that agents have a set of beliefs, desires and intentions (i.e., following a BDI system as the work examined above), but focus on the set of beliefs.

Finally, authors in [5] also discuss how they expand the logical language so that it includes implications, how CSs of agents are considered and how these are updated based on the moves the agents make during a dialogue. Particularly, the authors present the locutions they employ in their protocol, the rationality behind them, the available responses as well as the effect of these locutions in the CSs of participants.

2.2.6 Deliberation dialogue systems

In deliberation dialogues, participants need to jointly decide on an action or a course of action. Here, we briefly describe two important works related to this field. Later on in this chapter, applications with similar features are discussed.

In [130] the authors develop the first formal framework for deliberation dialogues called Deliberation Dialogue Framework (DDF). After presenting the characteristics that differentiate deliberation dialogues from other types of dialogue, the authors present a formal model of deliberation dialogue which consists of eight stages. To define these stages the authors describe some necessary features (types) such as *actions, goals, constraints, perspectives, facts* and *evaluations*. Notice that later, using a sentential language, sentences moved during the dialogue are instances of these types. The stages characterising deliberation dialogues are: *Open* (i.e., where the dialogue starts with a question regarding what is to be done), *Inform* (i.e., where agents discuss desirable goals, constraints on actions, evaluation of proposals and facts relevant to evaluation), *Propose* (i.e., where agents suggest possible actions-options), *Consider* (i.e., where agents comment on proposals), *Revise* (i.e., where agents revise goals, constraints, perspectives and actions-options), *Recommend* (i.e., where agents suggest an option for action and either they accept it or reject it), *Confirm* (i.e., where agents confirm the acceptance of their choices), and *Close* (i.e., where agents close the dialogue). The authors also specify that the aforementioned stages may occur in any order and participants can visit them as often as they desire, as long as they obey to some rules that the authors describe in the paper (see [130] for details).

Later on in [130], the locutions that enable DDF are presented. These are: *open_dialogue*, *enter_dialogue*, *propose*, *assert*, $prefer$, $ask\_justify$, *move*, $reject$, *retract*, *withdraw_dialogue*. Details on the contents of such locutions are also discussed, as well as the effects of these locutions in the CS of the participating agents (which are public, but only the participant's own utterances lead to additions into its CS). Of course, a protocol with rules on relationships between the locutions is also given (e.g., which locutions may be used as a response to *ask_justify*), but is considered relatively liberal. Additionally, the authors associate locutions to each stage given earlier. Finally, they evaluate DDF and the associated protocol by comparing it to human deliberation dialogues, considering their protocol from the perspective of the deliberation processes it implements and considering the outcomes, if any, that deliberation dialogues conducted under the DDF protocol achieve.

Another notable deliberation dialogue system is proposed in [112]. This work is based on [157] described earlier in Section 2.2.2. Thus, the deliberation system proposed in [112] provides, similarly, an explicit reply structure, a turntaking function and a termination rule, dialogical statuses of moves similar to the ones discussed in Section 2.2.2 (ensuring coherent dialogues), different protocol rules (which can be added/discarded depending on the domain) and an anytime outcome which can also be used to decide the winner of the dialogue at that particular moment. Note, arguments here are formed using inference trees of strict and defeasible rules, grounded on the formalism of arguments in [160].

The authors, however, had to make some modifications to the system presented in [157] so that their system accommodates deliberation dialogues. Firstly, more than two participants are allowed in [112]. Additionally, notions such as relevance and protocol rules had to be revised accordingly, and multiple proposals (instead of just one claim) are discussed during the dialogue. Moreover, the dialogue outcome is no longer a direct result of the moves. Finally, a winning function is needed to select a single action from all actions that are proposed, or possible none if there is no acceptable option.

3 Implementation building blocks

As noted above, the previous section covered the theoretical basis of computational argumentation and the work on argumentation-based dialogues that was built on that theoretical basis. In this section we now look at some of the software tools that have been developed to support argumentation-based dialogue. Many of these have been developed as implementations of that theoretical basis — for example, given the description of formal dialogues, it is clear that an implementation could benefit from a tool that computed the extensions of a set of arguments — and the mainstream of work progressed assuming that the route to argumentation-based dialogue software systems would always be through implementing formal models. However, more recently, the successes of natural language processing[6] has led to a second branch of work on argumentation-based dialogues, that which is based on NLP-derived chatbots. We therefore take a quick look at some of the work in that direction.

3.1 Components of an argumentation-based dialogue system

In considering the tools, it is helpful to identify some common components of argumentation-based dialogue system architectures, as shown in Figure 2. Taken together, these components represent a super-set of the components implemented in the set of applications discussed in this chapter. The diagram is meant to be neither prescriptive nor exhaustive. As the remainder of the chapter unfolds, the reader may find it helpful to refer back to this figure to understand the types of components implemented in the systems discussed and how they are placed within an overall schema.

Note that we separate the components into "Front End" and "Back End" elements. In programming, back-end is a commonly used term to describe the underlying infrastructure that drives the involved application. In the context of implementations of argumentation systems, reasoning engines, for example, fall into this category. Their task is to steer the decision-making processes, thus guiding the software towards its goal. We consider back end components in Section 3.2.

Computational argumentation is not only leveraged for back-end purposes. Argument graphs, for example, represent an informative way to display pieces of knowledge and the relations subsisting between those pieces. Such visual and interactive elements constitute (part of) the user

[6]The confluence of argumentation and natural language processing (NLP) has been long in the making, with the Argument Mining workshop in its 11th year, as of the time of writing, and the workshop on Computational Models of Natural Argument about to have its 24th instantiation.

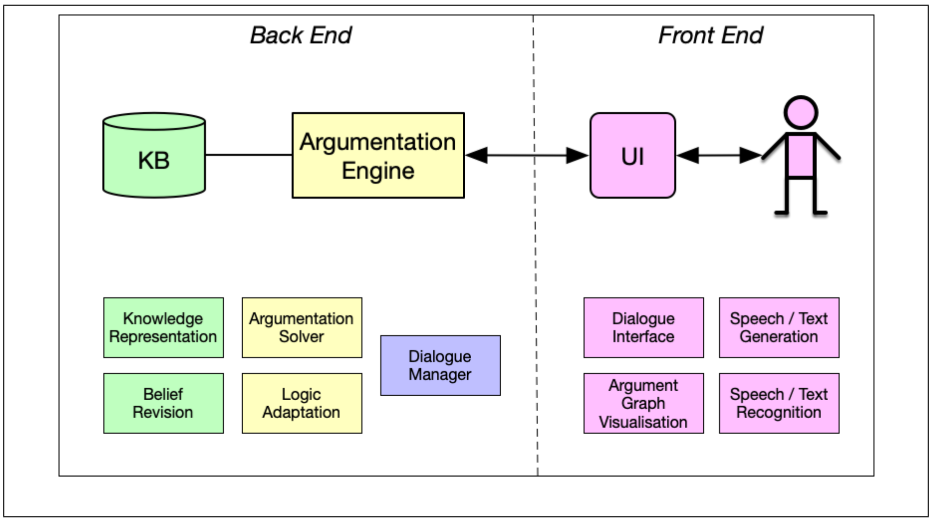


Figure 2: Architecture components. "Front End" and "Back End" separation is indicated.

interface, that is to say, the front-end component. For this reason, it is worth reviewing the existing argumentation-based dialogue applications that have both back and front-end argumentation-related components. We call these end-to-end argumentation components, and we consider these in Section 3.3.

3.2 Back-end argumentative implementations

One of the main purposes of computational argumentation is to enable the resolution of conflicting knowledge, thus allowing for a selection of the most appropriate (i.e., justified) pieces of information. *"A decision is a choice between competing beliefs about the world or between alternative courses of action. [...] Inference processes generate arguments for and against each candidate. Decision making then ranks and evaluates candidates based on the underlying arguments and selects one candidate as the final decision. Finally, the decision commits to a new belief about a situation, or an intention to act in a particular way."* [84]. Decision-making processes can be encoded as problems whose solutions are rendered by the computation and evaluation of AFs: an argumentation engine is essentially a reasoning tool driven by the same logic and process. The resulting acceptable entities provide a strong (logical) rationale for and against a given decision, while also leaving space for further deliberation [69]. Such an argumentative decision-making apparatus can be a useful addition to real-world software applications concerning defeasible reasoning, as advocated by the comprehensive study of Bryant and Krause [36]. Without any claims of completeness, we now provide a brief overview of one of the most common types of component of reasoning engines leveraged by argumentation-based dialogue applications: the solvers.

A specialized piece of software that encodes and provides the solution to a particular computational problem is known as a 'solver'. Popular stages where a plethora of different solvers for abstract argumentation decision procedures are presented are the ICCMA (International Competition on Computational Models of Argumentation) events [24, 86, 106, 118, 185]. In this competition, various pieces of software are evaluated according to their capabilities of addressing computational argumentation-related reasoning challenges in connection with specific σ semantics: for example, the enumeration of σ-extensions in the AF and the credulous and sceptical membership of a particular argument to at least one (credulous) or each (sceptical) σ-extensions. Among these computational argumentation solvers, we can acknowledge *AFGCN v2* [129] and *PYGLAF* [2], both of which harness Python scripts to achieve the desired results. In particular, *AFGCN v2* leverages an approximation method based on the Graph Convolutional Network (GCN), whereas *PYGLAF* combines Circumscription [133] and SAT solvers.

Similarly, SAT encodings and solvers are employed by *μ-toksia* [140] (either Glucose [9] or Cryptominisat [176]), *FUDGE* [184] (whose reduction-based method and sophisticated encodings ensure an optimized procedure over the benchmark) and *Crustabri* [117]. The latter stems from a rewriting of CoQuiAAS [116] developed using the Rust language. Other examples are *FARGO-LIMITED* [182], an approximate reasoning tool that relies on a variant of the standard DPLL search algorithm [23] and *HARPER++* [183], a solver whose operations hinge on the grounded semantics and its properties. It is also worth mentioning *ASPARTIX*-

V21 [74] and *ASPforABA* [122], both of which make use of Answer-Set-Progrmamming (ASP) encodings for, respectively, abstract and structured (ABA [72]) computational argumentation tasks. Leveraging a different approach, *ConArg* [25] takes advantage of Constraint Programming (CP) techniques and heuristics (via a specific C++ toolkit) to provide its output. Finally, *A-Folio DPDB* [82] resorts to an inventive solution by leveraging DPDB (i.e., a general framework designed to address counting tasks via dynamic programming and database management system [83]) adapted for computational argumentation reasoning purposes. We conclude the list by mentioning *AGNN* [63], an Argumentation Graph Neural Network that learns how to predict the likelihood of an argument being credulously and sceptically accepted.

3.3 End-to-end argumentation implementations

The work in the previous section largely consisted of implementations of formal systems. Here we start by considering end-to-end systems that are based on formal models before turning to chatbots.

3.3.1 Panoptic engines

Similarly to solvers, panoptic (or *all-encompassing*) engines are suites of different pieces of software that perform specific calculations concerning computational argumentation semantics. However, unlike standard solvers, those engines are designed to provide additional functionalities and customisation tools (e.g., knowledge base manipulation, domain selection, underlying logic adptation, graph visualization). Among these reasoning tools, we can include *ArguLab* [156] which computes (and graphically visualizes) the extensions of an AF, engages in structured dialectical exchanges to prove the acceptability of the justified arguments and incorporates considerations of judgement aggregations [45] to handle the stance of groups of agents.

Other examples are *Prengine* [101] and *PyArg* [34], both implemented in Python. The latter is a comprehensive tool capable of executing different computational argumentation tasks, including AFs (either abstract or structured) generations, evaluations and visualization. *Prengine* is instead designed as a multi-purpose engine that handles

Probabilistic Assumption-based Argumentation (PABA [73]) by translating Probabilistic Argumentation (PA) models into PABA, implementing inferences about arguments likelihood and computing their semantics. Even *NEXAS* [66] harnesses Python (in particular, the pandas library[7]) to provide an interactive exploration of the solution space, statistical analysis and a correlation matrix for the acceptance of individual arguments for the selected semantics.

On the other hand, *Argue tuProlog* [37] leverages a reasoning core Java-based Prolog to specify whether a claim can be argumentative and evaluates the outcome by tracing an argument game envisaged to prove such a claim. Another Prolog-implemented engine is *CaSAPI* [85], whose features include supporting users' customisation regarding argument, semantics and domain selection within the ABA framework. Furthermore, we can also acknowledge *IACAS* [189] as being one of the oldest prototypes of an argumentation engine whose purpose concerns the evaluation of arguments via two-party immediate response disputes. Finally, *ArgTrust v1.0* [180] is an argumentation engine implemented in Java, whose underlying methodology [179] reasons over data by assigning values to the arguments (and their relations) depending on how much the source is 'trusted'. A later version, *ArgTrust v2.0* [173], was implemented in Python and MySQL and facilitated an interface for users to interrogate the underlying AF.

3.3.2 Chatbots

Finally, we look at chatbots. These are conversational software systems designed to mimic human discourse mostly to enable automated online guidance and support [39], thus allowing humans to interact with digital devices as if they were communicating with a real person [146]. These computer programs generate responses based on given inputs producing replies via text or speech format [11, 175] employing different architectures [61]. Indeed, the long history of such conversational agents stems from rule-based, scripted template chatbots (e.g., the famous ELIZA [195]), whose replies are predefined and returned according to a series of NLP-encoded rules. The field has advanced towards retrieval-

[7] `https://pandas.pydata.org/`

based architectures (e.g. A.L.I.C.E. [191]), in which responses are instead pulled from a corpus of sentences according to the received input, and most recently centres on generative models, for example the well-known ChatGPT[8]. The generative architecture, which grants an agent the ability to formulate its own original responses rather than relying on existing text, hinges on the recent Transformer technology [188] that revolutionized the entire field of chatbot research[9]. Interactive agents engineered upon such a Transformer-based structure convey impressive performances within open-domain conversations (although they are not immune from various shortcomings [144]), while previous bot architectures could only aim at closed-domain conversations[10]. While chatbots are not, in general, argumentation-based, we mention them here because, as we will see, there have been recent efforts to develop argumentation-based chatbots.

Note that chatbots can be considered end-to-end software implementations[11] where an underlying response architecture elaborates the replies to be sent into a specifically designed chatbox. Here, the user will be able to interact and dialogue with the bot in text or speech format. This ability is attractive from a user interface point of view and is one of the reasons that chatbots are an interesting element of an argumentation-based dialogue system.

4 Selected applications

Having briefly clarified the background notions underpinning the whole chapter, we are ready to dive into a discussion of existing applications of argumentation-based dialogue. We do this according to dimensions described in Section 4.1 as well as components of the dialogue and the employed argumentation framework. Overall, we did not draw a strict line, and we opted for a comprehensive review by including all the per-

[8] `https://chat.openai.com`

[9] Arguably transformers have revolutionized the whole of NLP, as well as having found applications in related fields such as computer vision and genomics.

[10] A comprehensive survey of chatbot history can be found in the work of Adamopoulou and Moussiades [1], whereas a review of Transformer-based conversational models has been conducted in the study by Zhao et al. [201].

[11] Hence their inclusion here.

tinent research we could find. Our survey has not been constrained by dates of publications (although, where feasible, we preferred the latest version of a particular line of work), domains or evaluation method: we were only concerned with the application of argumentation-based dialogues, whether fully-fledged developed or just sketched, whether their structures relate to back, front or end-to-end operations, irrespective of the dialogue protocol, software tool or dataset (if any) adopted.

We start, in Section 4.1 by presenting ways to structure the literature on argumentation-based dialogue systems — see Tables 2 and 3, justifying the analysis and discussing some aspects of it. Then we proceed to examine individual systems. First, in Section 4.2 we describe an application, implemented in the health sector by a team inclduing many of the authors, as a use-case study. We do so because it represents a complete system of a dialogical application, comprising of all the components introduced in Figure 2. Thus, it represents a good example of the desired pieces and features of a fully-developed (in the sense of both back and front-end) dialogical application in the field of argumentation.

Next, we broaden the discussion under two main headings. First, in Section 4.3 we discuss systems that are built on theoretical models like those introduced in Section 2. Then, in Section 4.4 we look at work on chatbots, reflecting the more recent work that has grown out of research in the natural language processing community. In both of these latter sections, we draw on the distinction between dialogue types (see Section 4.1) as a way of structuring the discussion.

4.1 Methodology

In this section, we describe ways of structuring the current literature on applications of argumentation-based dialogues that we use in the remainder of the chapter. Tables 2 and 3 identify six different dimensions that we use as a basis for comparison: **application domain**, **user interface**, **dialogue type**, **data sets**, **software tools** and **evaluation**. The references cited in this table are discussed in detail in Sections 4 and 5. Here we limit ourselves to a few, more obvious remarks: that health applications dominate (though this is perhaps skewed by our work on CONSULT, see below); that persuasion and inquiry dominate in terms of the Walton/Krabbe classification; that much work is evaluated

Application domain	
health	[12, 49, 52, 55, 77, 109, 113, 139, 165, 200]
other	[6, 16, 21, 33, 46, 54, 95, 111, 121, 151, 163, 171]
Dialogue type	
inquiry	[21, 109, 139, 142, 151, 171, 200]
deliberation	[109, 139, 165]
persuasion	[6, 33, 46, 54, 55, 95, 96, 121, 163, 165, 171]
information-seeking	[96, 151, 165, 171]
negotiation	[16, 111]
Evaluation	
formal proofs	[33, 46, 81, 109, 121, 142, 151, 200]
human participants	[6, 54, 55, 111, 163]

Table 2: Ways to structure the literature on argumentation-based dialogues: Application domain, dialogue type and evaluation method.

formally, as one would expect from a literature with its roots in formal logic, but that an increasing number of papers include some kind of human-participant study; that there is no consensus on what software tools to use; that a large (and growing) number of systems make use of some form of chatbot, perhaps in response to the natural dialogic approach of argumentation; and few existing systems are data-driven despite the existence of a number of existing datasets.

The formal concepts of both abstract and structured argumentation frameworks, along with the notion of argumentative-based dialogue protocols, yield several software implementations that are reviewed in the following sections.

4.2 Consult: argumentation-based dialogue in decision support

We start by discussing an argumentation-based dialogue system which was developed by the authors of this paper as part of the CONSULT decision support system.

Software tools	
ASPARTIX	[49, 52]
ArgTrust v1.0	[171]
JADE	[200]
DGEP	[18, 21, 142, 174]
WFS	[139]
JaCaMo/Dial4JaCa	[76, 150]
Tweety	[109]
$Jack^{TM}$	[16]
programming languages only	[33, 46, 111]
User Interface (UI)	
chatbot	[6, 12, 49, 52, 54, 55, 91, 95, 96, 113, 163, 165, 171]
(other) use of NLP	[21, 76, 81, 142]
simple UI	[21, 33, 46, 109, 111, 121, 142, 200]
Data sets	
ACKTUS	[200]
National Service Centre E-Crime Dutch Police	[21, 142]
AIFdb	[18, 121, 174]
EDiC	[111]

Table 3: Ways to structure the literature on argumentation-based dialogues: Software tools, user interface, and data sets.

Decision support systems (DSS) represent valuable tools that assist human users in making well-informed choices via the provision of pertinent recommendations. In the healthcare sector, such DSS prove to be especially useful for a number of reasons, including patient safety, cost containment and improved quality of documentation [178]. Indeed, there exists a long history of expert systems in the medical domain field [164] that can be traced back to MYCIN [168]. *Clinical decision support systems (cDSS)* are mostly characterised by machine learning approaches, although the literature also comprises a number of cDSS driven by computational argumentation as the underlying reasoning procedure [58, 64, 115, 143]. Surely, as highlighted by Lindgren et al. [125], this is a thriving area for argumentation since it can handle the conflict of knowledge occurring when multiple stakeholders are solicited with regard to specific medical cases.

In particular, CONSULT [12, 77, 113] is a data-driven cDSS that leverages an argumentation reasoning engine to help patients manage their conditions in collaboration with healthcare professionals[12]. The system receives multiple inputs (coming from different wearable wellness sensors, clinical guidelines, the patient's preferences and electronic health record), which then encodes and structures as arguments. The reasoning engine runs on the ASPARTIX solver [75] and computes recommendations by instantiating textual explanation templates with acceptable (according to Dung's semantics) arguments [114]. The outcome of this operation is stored in an internal repository of the cDSS whose elements feed the EQRbot, the chatbot responsible for interacting with the patient [52, 49].

Drawing from a previous dialogue framework [165], the EQRbot engages in an Explanation-Question-Response (EQR, hence the name of the chatbot) dialogue starting from the instantiations of the homonym argument scheme, embedding the initial CONSULT recommendation, and then proceeding by clarifying any additional follow-up user question. The implementation of the dialogue presented in [49] is still limited and may be extended in the future by including the full spectrum of available locutions of an Explanation-Question-Response dialogue (originally sketched in [132]) according to the formalization of [47, 51]. The advantages of such a protocol comprise the following. First, a simple design that avoids meta-level locutions to manage the dialectical interplay whilst conveniently embedding multiple dialogue types. Compared to other dialogues that require a Control Layer, the simplicity of the EQR design favours its implementation. Second, EQR exchanges of arguments result in interactions satisfying desirable properties of explanations (i.e., exhaustivity, selectivity, transfer of understanding and contextuality). Lastly, the information conveyed by a terminated EQR dialogue proves to be justified by a number of compelling reasons. In-

[12]The overall (microservice) architecture of CONSULT can be found in [57].

deed, such an explanation produces sound and complete results with respect to Dung's AFs admissible semantics, thus allowing evaluation of the EQR dialogue moves using any proof theory, algorithmic procedures or methodologies semantically associated with computational argumentation.

The next two sub-sections contain more examples.

4.3 Applications based on formal models

This section reviews implementations of argumentation-based dialogues based on the theory summarized in Section 2. We start by covering software tools that can be used to implement dialogues and then move to look at individual applications. We structure this latter part of the current section using the Walton and Krabbe typology.

4.3.1 Tools for implementing a dialogue

The DGEP (Dialogue Game Execution Platform) [18] is a system capable of interpreting dialogue game specifications which are expressed in an amended version of the Dialogue Game Description Language (DGDL) [196] named DGDL+[13]. Based on these specifications, DGEP[14] generates dialogue templates [22], which are schematic representations of individual moves in a dialogue, along with their replies and connections to the underlying argument structure using the Argument Interchange Format (AIF) ontology [59]. The AIF serves as an abstract core ontology for representing different theoretical and practical approaches to argumentation, acting as an interlingua between various argumentation approaches and enabling evaluation of arguments constructed in visualization packages using different argumentation theoretic semantics [20]. The AIF also underlies the Argument Web [19], a linked data Semantic Web structure containing numerous claims and arguments with different relations between them. Thanks to DGEP, dialogue histories with explicit reply structures can be formed by combining multiple templates,

[13]The DGDL+ specification and example dialogue protocols can be found at `https://www.arg.tech/index.php/research/dgdl/`.

[14]DGEP is available at `https://github.com/arg-tech`.

allowing existing argument structures in the Argument Web to be navigated and updated using dialogues.

DGDL+ includes several requirements as inbuilt predicates, such as CS checks and role checks, and it provides a general-purpose predicate for indicating arbitrary functions not defined in the protocol specification. DGEP processes DGDL+ specifications as if it were a compiler, converting them into an Abstract Syntax Tree (AST), and then further transforming it into a Python data structure representing the hierarchy of elements within the DGDL+ specification. The main goal of DGEP is to execute dialogue games specified in DGDL+, building AIF graphs and expanding the Argument Web. It develops the legal move list for each participant and handles the instantiation of rule effects and other parameters, like initial states, allowing agents to use the Argument Web as their KB. Every turn, DGEP generates all legal moves and delivers them to participants in the form of a 4-tuple, consisting of a moveID, opener (informal indication of the utterance), reply (formal structure of the move), and a fragment of AIF corresponding to the move. This fragment provides information about the structure of the move, allowing agents to create queries on AIF and extract relevant arguments. However, dialogue strategies for artificial agents are not fully implemented, so move selection is random. Finally, DGEP updates AIF structures during move execution based on theoretical accounts discussed in [22]. Note, a set of simple web service interfaces is provided which allows clients for both autonomous agents and human interfaces to connect and play instances of dialogue games.

DGEP is also the core of the modular architecture called Dialogue Utterance Generator[15] (DUG) introduced in [174]. DUG finds propositional content to instantiate abstract move types, provided by DGEP, into concrete moves. Specifically, DUG uses content descriptors and associated content locators. Content descriptors describe how variables in the "reply" object should be populated, whereas a content locator provides content by querying a MySQL database. As a result, the reply is given to a participant of the dialogue as a concrete legal move they can make. If the result outputs multiple values, then a concrete move is created for each piece of content. AIFdb [119], argument mining [120],

[15] DUG is available at `https://github.com/arg-tech`.

logical representations or other queriable sources may be considered for content instantiation.

4.3.2 Persuasion

Following the Walton and Krabbe typology and mirroring the previously introduced formal models of dialogues, this subsection discusses three systems for persuasion.

First we look at Polemicist [121]. Polemicist[16] is a dialogical interface for exploring complex debates from the BBC Radio 4 programme The Moral Maze. It allows for the user to interact with software agents, who act as the participants in the original programme, and explore the topic as they wish, asking questions to delve into the areas they want. The agents' KBs are extracted from analysis of the original episodes (represented in the AIF [59]).

Polemicist translates navigation of the generated knowledge graph into a series of dialogical moves conducted according to the dialogue game for persuasion from [158]. It uses a fixed protocol, defined in DGDL [18], where the user is the moderator of the debate, allowing agents to select topics and control the flow of the dialogue. Its interface contains two panels, where the one lists the participants with green and red highlighting showing their agreement or disagreement, respectively, with the most recent point made. The other depicts the history of the dialogue as well as a sub-panel which enables the user to either ask the opinion of a participant or question the reasons why the participant's opinion holds. Note that the dialogue history allows the user not only to view the dialogue but also return to previous points, and listen to the original audio associated with each text segment. Finally, it is worth mentioning that the Polemicist relies on pre-annotated material from AIFdb[17] [119] to provide the responses of the software agents in a dialogue.

Next we discuss DISCO [33]. DISCO, or, more correctly DIScussion COmputation[18] provides a web-based implementation of the Preferred

[16]Polemicist and other argumentation-related work can be found in `http://www.johnlawrence.net/projects.php`

[17]`http://www.aifdb.org/`

[18]DISCO can be found at `http://disco.cs.cf.ac.uk`

[43] and Grounded Discussion Game [41]. These two models [41, 43] are theoretical models of persuasion that build on the earlier work of [137]. DISCO is written in Javascript, and all computation is performed on the client side. The motivation of the authors is to implement these discussion games for the purpose of explanation. The user can choose to open an existing AF (in a JSON file format) or construct one manually by adding arguments and attacks to an initially empty canvas, so that they can play either the preferred or grounded discussion game. The user may choose to play as proponent or opponent and accordingly they take turns in moving arguments (where a move by the user can be typed into a text field, or can be selected by clicking on the relevant arguments). For both games, if the computer has the role of the proponent, it will win the game as it follows the associated winning strategy [42]. Finally, DISCO provides additional features such as saving the AF (possibly as an image), allowing the user to ask for recommendations regarding the moves they can make and viewing the grounded labelling and associated min-max numbering for the Grounded Discussion Game.

Finally, we look at *Argument-Based Discussion using $ASPIC^-$* [46]. This is a variation of DISCO[19] where the construction of arguments is based on the $ASPIC^-$ framework, a variant of $ASPIC^+$ where the definition of attack is more suitable for interactive applications [44]. Specifically, rule-based arguments are constructed from an underlying KB, stored in a text file, instead of abstract arguments. The demonstrator is written in Python3, it does not require any non-standard libraries, and has been tested to work under both Windows and Linux. Firstly, propositions and strict rules are specified in the file, followed by defeasible rules where those in the same block have the same strength and those in later blocks have a higher strength. Notice that defeasible rules have names for undercutting. To start the application, one starts from the command line adding as parameters $-wl$ (for weakest link principle) or $-ll$ (for last link principle) or $-do$ (for democratic order) or $-eo$ (for elitist order) [138]. After this step, a query to the inference engine is made regarding whether a statement is justified or not (i.e., if the statement is the conclusion of an argument in the grounded extension). After the engine's response, the user may ask for an explanation and start a

[19] ABDA can be found at `https://github.com/Schirmi136/ABDA`.

discussion with the system. If the statement is justified, the system will assume the role of the proponent and the user the role of the opponent, otherwise the roles change correspondingly. At the moment, arguments played in the game are written in a nested, machine readable way and the target is for these to be given in natural language. The authors motivate the use of natural language by briefly suggesting the medical domain as an example for the implementation of their application.

4.3.3 Inquiry

Next we consider three argumentation-based dialogue systems that implement inquiry dialogues.

We start with [200], where the authors present a multi-agent framework designed to handle uncertain or inconsistent information in a distributed environment, and implemented in a clinical decision-support system (called DMSS-W) for diagnosing dementia. Specifically, the system involves a dialogue between a novice physician (PA) and a medical domain expert (DA), which is represented by the system, where the PA suggests a hypothetical diagnosis in a patient case. This is verified through the dialogue if sufficient patient information is present, otherwise the user is informed about the missing information and potential inconsistencies in the information as a way to support their medical education. Notice that pragmatic evaluation is left as part of future work and planned to take place in clinical practice. The framework builds upon the inquiry dialogue introduced by Black and Hunter in [28], allowing agents to collaborate in finding the best solutions and new knowledge while also addressing inconsistencies. The dialogue system consists of two participating agents, moves (which consist of the sender, the move type, the dialogue type, and the topic of the dialogue), a protocol that defines legal moves and other aspects of the dialogue system (such as its outcome), as well as a dialogue history (see [200] for details on the possible values of these components). It incorporates possibilistic logic [70] to build an argumentation system which captures uncertain information and degrees of confidence in knowledge sources.

Practically, the system utilizes data from a platform called ACKTUS, a web-based tool for modeling medical knowledge into rules and claims in natural language. The multi-agent system (MAS) is developed using the Java Agent Development Framework (JADE). By using JADE, the authors can implement DA and PA as agents of MAS as well as define the components the agents have access to (e.g., their KBs and the arguments they can instantiate) and generate dialogues between them. Thus, JADE acts as the inference engine of DMSS-W. The dialogue between PA and DA leads to a diagnostic result, which is presented in the DMSS-W user interface. The domain experts model interaction objects (IOs) through ACKTUS and store them in the domain repository. Each IO contains scales with different values to determine the level of certainty. The user answers questions in the interface by clicking on scale values, which are used as state beliefs in reasoning. Rules are created based on premises, conclusions are derived from the IOs, and possibilistic values are assigned to these rules. As the reasoning process may lead to conflicting arguments due to uncertain and inconsistent data in the KBs, two different strategies are introduced so that the user may choose which one they will use to manage the conflict.

The second system we consider is that of [139]. Here, a cooperative layer within a multi-agent system is presented, focusing on a scenario involving an older adult's needs and preferences for support in daily activities within a smart home environment. The agents in the system need to find optimal actions despite partial and inconsistent information, considering the changing needs and wishes of the older adult. The argumentation dialogues in this system are again inspired by [28] (although more participants are allowed to participate, and the components of a move are slightly different, resulting to changes in the CSs of agents too) and use default theories (extended logic programs) that can be mapped into Assumption-Based Argumentation (ABA) [32, 80] for dialogue inference. The intelligent infrastructure, called As-A-Pal, includes three instantiated agents: Environment Agent, Activity Agent, and Coach Agent. These agents, which possess rule-based KBs, collaborate to provide support to the older adult in conducting activities. Specifically, deliberation dialogues occur when they attempt to agree on actions to perform in certain situations, and agreement rules are used to reach a consensus. The system utilizes *Well-Founded Semantics (WFS)* [71] as a reasoning engine[20] to infer information from logic programs. Inquiry

[20]Details can be found at `https://github.com/esteban-g/wfsargengine`.

dialogues [28] are applied to validate the truth of “agreement atoms” or agreement rules. If an agreement atom holds true in a given state, it represents a particular belief’s truth in the entire As-A-Pal system.

Finally, we consider the work of Bex and colleagues [21, 142]. In [21], an initial sketch is given for an artificial agent handling the intake of internet trade fraud by combining natural language processing with symbolic techniques for reasoning about crime reports. The system serves two main types of users: complainants filing new criminal complaints and the police who want to analyse reports and build case files. Both interact with the system through the dialogue interface, which allows them to submit input and view the status of the dialogue, including open questions. A dialogue manager, based on the Dialogue Game Execution Platform (DGEP)[21] [18], specifies the dialogue protocol, such as turn-taking and legal moves, and keeps tracks of users’ commitments. Multiple scenario reasoning agents can participate in a dialogue, using predefined fraud schemes from a library and crime report repository. These agents can match scenarios to typical fraud schemes, compare scenarios based on available evidence, and elicit further information from users. Determining the true scenario often requires additional evidence, turning the investigation into a process of inference to find the best explanation.

Later in [142] [22], the subsequent development of the intake agent is regarded as argument-based inquiry dialogue, once more inspired by [28]. $ASPIC^+$ is used to define an argumentation system where defeasible rules represent the laws and practices surrounding trade fraud are combined with the citizen’s knowledge of the specific situation they observed, to build arguments for and against the main claim made by the citizen. Additionally, natural language processing techniques are used to extract automatically the initial observations from free-text user input [167], so that these observations can be combined with rules concerning trade fraud in the argumentation setting to build arguments for and against the claim “fraud”. The notion of *Stability* is also discussed (from a theoretical point of view) which is used to decide whether the addition of

[21] See more details on DGEP in the next Section

[22] The Dutch Police’s website, which implements the intake agent (in the Dutch language), can be found at `https://aangifte.politie.nl/iaai-preintake`.

more observations from the citizen in the future can change the acceptability status of the "fraud" claim. If not, the dialogue terminates; otherwise a question policy component finds the best question to ask given current observations. In other words, the stability component provides a termination criterion that prevents the agent from asking unnecessary questions. Notice that the stability component can also be perceived as a "tool" for dealing with enthymemes, since it essentially signifies whether or not an argument is complete. See [141] for further insights and applications of the intake (dialogue) agent, which also includes an empirical evaluation and provides a deeper comprehension on how it captures enthymemes that miss some of the necessary support to entail a conclusion.

4.3.4 Information-seeking

From the applications of information-seeking in the literature we pick that of [151] to examine in detail.[23] In this paper, an argumentation framework is described where agents are able to exchange shorter messages when engaging in dialogues by omitting information that is common knowledge. These messages are treated as enthymemes; and shared argumentation schemes are used, as well as common organisational knowledge, to build an enthymeme-based communication framework. Concerning the argumentation schemes, the "Argument from Position to Know" from [193] with associated critical questions are applied, but referring to organisational concepts. According to the authors, such argumentation schemes can be represented in structured argumentation, using defeasible inferences. Additionally, they use first-order logic to represent arguments, arguing that this is a reasonable choice given that most agent-oriented programming languages are based on logic programming. The authors also argue that instantiating arguments from argumentation schemes allows agents to use such arguments for both reasoning and communication processes, and so they use argument schemes to guide the decoding of enthymemes into the original sender's argument.

Note that all agents/participants are aware of other agents' roles in the organisation as well as the associated features/abilities related to

[23] Note that the same implementation can be used to model inquiry dialogues.

them. Furthermore, the authors [151] show that their work addresses some of Grice's maxims, proving that agents can be brief in communication, without any loss in the content of the intended arguments. To implement this enthymeme-based communication[24] framework the JaCaMo Platform [31] was employed. Finally, to evaluate their framework, the authors use scenarios of argumentation-based dialogues that use different argumentation schemes and argumentation-based protocols from the literature. At first the information-seeking protocol specified in [148] was used, but later on the inquiry protocol specified in [149] was also employed. However, in both scenarios, the same locutions are leveraged, which are described within [151] together with their effects on the CSs of the agents and the dialogue.

4.3.5 Negotiation

Representing negotiation, we have [16] and [111]. In [16], the authors propose a formal description and implementation of a negotiation protocol between autonomous agents using persuasive argumentation. In persuasive negotiation, an agent is trying to influence the behaviour of another agent using arguments supporting the proposed offers[25]. The logical language the authors use comprises of propositional Horn clauses, i.e. disjunction of literals with at most one positive literal which can be also written as implications. An argument is a a pair $A = (H, h)$ where h is a formula of a propositional language L, and H is a set of formulae of L such that H is consistent, $H \vdash h$ and H is minimal, similar to [5]. The KBs of the participants however are assumed to be consistent but attacks between arguments are defined as in [5], i.e. an argument A attacks another argument B on its premise(s). Additionally, a commitment store is used to track the arguments that have been publicly exchanged. Notice that agents can reason about trust and use trustworthiness to decide, in some cases, about the acceptance of arguments.

In regards to formalising their protocol, the authors use small computational *dialogue games*, i.e. a logical rule indicating that if Ag_1 per-

[24] The implementation of the framework is available open source in `https://github.com/AlisonPanisson/EBCF`.

[25] As a result, one might therefore consider this to be a form of persuasion dialogue as well.

forms action Act_1, and a formula of the logical language is satisfied, then Ag_2 will perform Action Act_2 afterwards. Five types of dialogue games are considered, namely: *entry, defence, challenge, justification and attack*, where the entry game allows agents to open the dialogue, and the rest represent the *chaining games* which constitute the main negotiation process between the agents. The protocol terminates either by a final acceptance or by a refusal of the proposal discussed. The locutions and moves that the agents can use depend on the dialogue game played. Moreover, different properties of the protocol are proved, and discussion over the complexity efficiency of the protocol takes place.

Finally, the authors describe the implemented prototype of their system, where they use the $Jack^{TM}$ platform [8]. $Jack^{TM}$ is an agent-oriented language based on the Belief–Desire-Intention (BDI) model, and offering a framework for multi-agent system development. It supports Java and includes all components of Java offering specific extensions to implement agents' behaviours, including support for logical variables and cursors which is helpful when querying the state of an agent's beliefs. In addition, both the agents and their KBs are implemented using $Jack^{TM}$, where the agents communicate with $MessageEvents$ representing actions that an agent applies to a commitment or to its content. A dialogue game is implemented as a set of events and plans, where a plan describes a sequence of actions that an agent can perform when an event occurs. An agent Ag_1 starts a dialogue game by generating an event and by sending it to the addressee Ag_2, then Ag_2 executes the plan corresponding to the received event and answers by generating another event and by sending it back to Ag_1, and so on. Note, to start the entry game, an agent chooses a goal that it tries to achieve which is to persuade its interlocutor that a given propositional formula is true. This is why a BDI event is used as it models goal-directed behaviour in agents, rather than plan-directed behaviours.

In [111], the authors describe a computational model of agreement negotiation processes, which involves natural reasoning. The general type of interaction the authors deal with represents a kind of directive interaction where the goal of one participant, Ag_1, is to get another participant, Ag_2, to carry out a certain action D. One of the authors' aims is to investigate actual dialogues and this is why they selected to analyse, three sub-corpora of the Estonian Dialogue Corpus (EDiC) which although includes mainly information-seeking dialogues, typical sequences of dialogue acts (DAs) were found in human–human spoken dialogues that form agreement negotiations and reflect reasoning of the participant who has to make a decision about an action. Their model is implemented in an experimental dialogue system as an application where a user participates in communication training sessions.

The application is implemented in Java supporting (text-based) interaction with a user in Estonian and employs only predefined set of sentences which they can select from a menu. The sentences are only classified semantically according to their possible functions and contributions in a dialogue (e.g., the sentences leveraged by Ag_1 to increase the usefulness of the action, the sentences harnessed by Ag_2 to indicate harmfulness of the action, etc.). These sentences are dealt as arguments, and private and public information are considered in each information state of a conversational agent. The private information of an agent Ag_1 contains: a model of their interlocutor Ag_2, a reasoning procedure[26] which Ag_1 is trying to trigger in Ag_2 to persuade them positively for the decision D, aspects of D under consideration, a set of DAs (including the proposal and statements for increasing or decreasing weights of different aspects of D for Ag_2), and a set of utterances for increasing or decreasing the weights (i.e., arguments for/against). Every utterance can be chosen only once by Ag_1 and so Ag_1 has to abandon its initial goal if there are no more arguments to move. The shared part of information contains a set of reasoning models, a set of tactics (such as enticing, persuading and threatening) and a dialogue history, i.e. the utterances together with participants' signs and DAs. Furthermore, update rules used for transitioning from an information state into another are also defined. However, notice that the usual aspects considered in this chapter, such as instantiating arguments via logical language or traditional protocol representation, do not take place in this work. Finally, an evaluation occurs where a group of volunteers used the application, and a user needs to accept to do D, but 65% of the dialogues did not have this result.

[26]The reasoning model of an agent in [111] is analogous to a BDI model, but more kinds of motivational inputs are considered for creating the intention of an action in an actor in order to understand the effects that these factors –namely *wish, needed, must*– will have on the reasoning process.

4.3.6 Deliberation

Here we discuss [109], which primarily handles deliberation dialogues, though it can also support inquiry. [109] presents the implementation of the DiArg argumentation-based dialogue engine. It focuses on automating sequential argumentation, i.e. the iterative resolution of sequences of *AF*s (mainly for deliberation but, as previously stated, also for inquiry dialogues). By resolution, the authors mean that extensions of an *AF* are determined where one is selected as the *AF*'s conclusion, either automatically or manually by a human user. Specifically, DiArg resolves abstract *AF*s. In DiArg dialogues, an *AF* sequence is created by expanding an initial *AF*, i.e. by adding new arguments and attack relations to it (and again resolved, and so forth). DiArg can also ensure that results derived from an *AF* sequence preserve *Reference independence* and *Cautious monotony* principles.

In software terms, DiArg is an open-source Java library[27], where the program code and its documentation allow for inspection of the underlying data structures and algorithms. DiArg also utilises Tweety [181], that provides Java libraries to define and resolve different types of formal argumentation frameworks, to implement argumentation-based dialogue systems. A scenario of a digital assistant for stress management [90] is described in the paper, where the assistant recommends stress-relieving activities (in the form of arguments) to a user who can then either accept the suggestion of the system and add it to their schedule, or reject the activity by attacking it with an argument. Finally, the authors discuss limitations of DiArg that relate to context support, integration with recommend systems approaches and interoperability enhancements in alignment with the AIF [59].

4.3.7 Other[28]

The work presented by Fazzinga et al. engineered a privacy-preserving dialogue system based on computational modes of arguments [81]. This

[27] The DiArg reasoner as well as an implemented dialogue example is available at `https://github.com/Interactive-Intelligent-Systems/diarg`

[28] As Dawkins notes in “The Selfish Gene” [67], any attempt at a taxonomy other than that based on evolutionary history will end up with a “miscellaneous” category. This is ours.

architecture focuses on data protection and explainability to address the mistrust that current dialogue systems can raise in their users. By means of an *Argumentation Module*, it is possible to probe the rationales behind the dialogue system responses and understand the supporting and conflicting reasons underpinning them. A Covid-19 vaccination case study illustrates how such an architecture can fit a real-world scenario. The system has also been formally evaluated by proving specific formal properties (such as consistency, well-formedness, and termination).

Also in this category is the Multi-Agent Intentional Dialogue System (MAIDS) framework. (Arguably this could appear in several of the previous sections since it supports peruasion, information-seeking and inquiry dialogues.) This combines argumentation theories with other features to support complex dialogue [76]. Several agents are instantiated and each provides unique expertise in the system. The *assistant* engages in argumentation-based reasoning (following the approach developed in [152]) the results of which are then translated into natural language and conveyed to the human user by the *communication expert(s)*. *Ontology expert(s)* handle various ontologies (e.g., OWL), whereas *domain* agents address the specificity of different domain applications.

4.4 Argumentation-based conversational agents

As previously stated, chatbots are interactive pieces of software with a specific history and recognizable features: a virtual chatbox (or log, especially for speech-to-speech agents) and a strategy to provide messages. Given their well-defined structure and characteristics, which further diversify according to the internal architecture and the operational domains, we choose to dedicate a separate section to examine the combination of such conversational agents with argumentative dialogues. While chatbots grew out of work on natural language processing, they may handle and deliver their responses by leveraging the protocols and the formalism of argumentation-based dialogues. Harnessing the dialogue logic, the conversational agent can optimize its strategy and move only the arguments that prove to be necessary for achieving its final goal[29]. We discuss work on what we might call "argumentation-enabled

[29]Some of the authors have recently written an extensive review of argumentation-based chatbots [50]. We invite interested readers to refer to such a study for a

chatbots" using the same structure based on dialogue-type that we used in Section 4.3. However, it is interesting to notice how all the reviewed works concern persuasion protocols or a mixture of dialogues that include persuasion.

4.4.1 Persuasion

As a first example, we could examine the work introduced by Hadoux et al. [95], which expands upon previous studies from the same authors [94, 103, 104], and depicts an overall framework for modelling beliefs and concerns in a persuasion dialogue. An implementation of such a framework is then envisaged via an *automated persuasion system (APS)*, a software application aiming at convincing the interacting agent to accept some argument. Following the asymmetric persuasion dialogue protocol illustrated therein (i.e., unlike the system, the user is restricted in choosing replies among the provided options), the proposed chatbot proves to be capable of identifying, within its knowledge base embedded in an argument graph, the most appropriate argument to posit. Essentially, the APS performs a Monte Carlo Tree Search coupled with a reward function to maximize the addressing of concerns (paired with the arguments of the graph) and the user's beliefs.

Similarly, the bot presented in [54] aims at persuading the interlocutor via a free-text interaction where the user's inputs are matched (by vector rendering and cosine similarity) with the (crowdsourced) arguments of the graph representing the knowledge base. The chatbot trains a classifier to detect the most common concerns of the persuadee and employs it to select counterarguments that will produce a result more compelling than a random choice. If no argument similarity is detected, then the conversational agent will resort to a default reply based on the user's concerns. Furthermore, the same authors presented an analogous architecture for a persuasion bot with the addition of a particular concern-argument graph [55]. By incorporating the knowledge base within such a small graph, it can be proved that no large amount of data is needed to generate effective persuasive dialogues. Interestingly, a preliminary analysis of the impact (appeal) of arguments addressing

detailed list of conversational agents employing computational argumentation beyond dialectical delivery.

the users' concerns in a persuasion dialogue performed by a chatbot has also been conducted in a dedicated investigation [56]. A different example of such a concern-based approach may be represented by Argumate, a chatbot designed to facilitate students' production of persuasive statements [91]. To provide appropriate suggestions, the bot retrieves its replies from an underlying argument graph, whose edges denote attack and support relations, via a concern identification method. Notice that the interactions between Argumate and the users occur both by typing and selecting predefined options.

A common trait amongst most of the above argumentation-based conversational agents is that, although the corpus from which they extract their replies is organized as an argument graph, there is no interest in any particular acceptability semantics [71]. That is to say, the knowledge base is organized and considered as a plain AF, where arguments and attacks are the only relevant features. In addition, most of these studies also account for a baseline chatbot which exploits a random strategy for selecting counterarguments from the available choices within the underlying knowledge base. The reason for this is to provide a means for comparing the developed bots which employ more fine-grained strategies for choosing their replies.

Another conversational agent that focuses on the delivery of persuasion dialogues is the chatbot designed by Andrews et al. [6]. Implemented by harnessing the AIML markup language [190], the bot comprises a planning component that searches over an argumentation model for the optimal dialectical path to pursue in order to persuade the user. The agent records the user's beliefs and updates this information whenever its interlocutor agrees/disagrees during the interaction. Such beliefs-revisions play an important role in the strategic planning of the chatbot.

Finally, one last chatbot (SPA), envisaged in the study of Rosenfeld and Kraus [163], employs an AF as the basis of a reasoning procedure to perform persuasive interactions. In particular, it embeds its knowledge base into a Weighted Bipolar AF (WBAF) and computes the argument that maximizes the framework evaluation function according to the user input. The score returned by the valuation function represents the reasoner's ability to support that argument and defend it against potential attacks. The dialectical interaction with the user follows a strategical persuasion dialogue protocol (optimized via Monte Carlo Planning [169]) that might involve updating the argument frameworks of both the persuader and the persuadee.

4.4.2 Information-seeking and Inquiry

As noted above, all the chatbots that we cover have some element of persuasion. Here we consider those which have some non-persuasion element. First, we consider the conversational agent implemented by Sassoon et al. [165], within the context of explanation for wellness consultation. This exploits deliberation and information seeking protocols, in addition to persuasion whilst exchanging instantiations of acceptable argument schemes with its interlocutor. The adoption of diversified dialogue protocols (i.e., persuasion, inquiry and information seeking) also characterises the chatbot-equipped robot proposed by Sklar and Azhar [171]. Retrieving the most appropriate argument constructed from its beliefs, an operation facilitated by the restricted options available to the user, the robot communicates with its human interlocutor in order to strategize about a treasure-hunting game and explain the rationale behind its decisions.

Finally, we consider the bot introduced in [96]. This German-language conversational agent, following the formalisation of [94], makes use of an argument graph to encode its knowledge base from which it retrieves main stances and counterarguments to engage the users in discussions concerning the ethical challenges of AI implementations. The delivery strategy is somehow ambiguous but seems to balance a mixture of persuasion and information-seeking, according to the specific stage of the conversation.

4.4.3 Evaluation of chatbots

The argumentation-based chatbots described above have typically been evaluated via specifically designed user studies. Since this differs from the way that much work on argumentation-based dialogue is assessed, we think it worth discussing in detail.

The SPA conversational agent introduced in [163] outplayed the baseline chatbot (which harnessed a different heuristic strategy) when tested in its persuasion task, thus proving capable of delivering human-like conversations. Similarly outperforming the baseline agent is the bot presented by Chalaguine et al. [56]. Indeed, the paper includes an experiment that shows how such a chatbot, by positing arguments that address the users' concerns, is more likely to positively change the users' attitude in comparison with another agent that does not employ such a strategy. An analogous interest in users' concerns is encompassed in a study implemented by the same authors [54]. The results (conjointly supported by the experiments in [94] and confirmed by [95]) conclude that a strategic chatbot accounting for concerns is more likely to provide relevant and cogent arguments.

Moreover, it is also worth mentioning the evaluation outcome of the other two aforementioned persuasive agents presented [6, 55]. The former bot provides fluent conversations with its interlocutors performing generally better than a purely task-oriented system. The latter, instead, shows how an interactive chatbot yields more compelling information than a static webpage. Resorting to pre- and post-dialogue Likert-scale questionnaires is the preferred evaluation choice of the work presented in [96]. The results record successful shifts of the opinions of 40-50% of the participants after engaging with the chatbot. Overall, the users acknowledged the quality of the arguments and the design of the conversational system. Lastly, the dialectical agent designed in [171], implemented and evaluated on a robot in [10], was further investigated in [172], where discussions conducted within the previous user study [10] were evaluated from the viewpoint of explanations provided. The results show how leveraging argumentation-based dialogue improves system performance and users' satisfaction, although no particular correlation was detected between these metrics and the possibility of receiving explanations.

5 Discussion

Just as the exchange of arguments influences our reasoning [134], so the engagement in dialogues considerably affects human lives in a plethora of different scenarios. Argumentation-based dialogues formalise inter-agent communication protocols and strategies, and their applications are likewise broad in scope and modalities. Whether chatbots, recommender systems, end-to-end software or just blueprints of future implementations, the literature reviewed highlights some common patterns that can be harnessed to underpin the following analysis.

Reading through our survey, it is clear that persuasion is the type of argumentation-based dialogue protocol that is most embedded in interactive software architecture, such as chatbots or cDSS (e.g., [6, 33, 46, 54, 55, 95, 96, 121, 163, 165, 171]). This is rather natural since argumentation-based formalisms prove to be quite effective in providing compelling strategies and replies to induce belief change, as suggested by the results of several studies [6, 54, 55, 56, 94, 95, 96, 163]. Another trend that emerges from our survey is the connection between eXplainable AI, argumentation-based dialogues and their applications. Indeed, providing clarifications about the inner workings of black box algorithms seems to be a thriving area of application for dialectical protocols that involve argumentation[30] [65, 187]. In particular, a frequent procedure to reveal the underpinning rationales of AI systems' decisions consists of retrieving acceptable information (from the pertinent knowledge base) according to specific argumentation semantics [49, 81, 165].

Although it is persuasion that has been mostly considered in dialogical applications for argumentation, there are works that investigate the implementation of other types of dialogues, too. Inquiry is an example of a dialogue type that has been studied several times as a practical application [21, 109, 139, 142, 150, 171, 200]. The cooperative nature of inquiry allows agents to combine their knowledge to find the truth regarding the matter discussed, and this is why it has been found useful in applications concerning various domains, such as healthcare [109, 139, 200], fraud investigation [21, 142] and communication in organisations [151] as well as human-robot teams [171]. As a side note, we observe that most of the applications we have found, are concerned with the healthcare domain[31] [12, 77, 49, 52, 55, 109, 113, 139, 165, 200], thus stressing the importance that efficient communication tools (such as argumentation-based dialogues) assume within the medical context.

[30]Doubtless this popularity is a result of the recent interest in eXplainable AI and its link with computational models of arguments [65, 132, 172, 187].

[31]This holds even taking into account the biases we introduced by describing multiple aspects of the CONSULT system.

Information-seeking and deliberation only appear to exist in applications that include more than one dialogue type, where persuasion or inquiry take precedence (e.g., [171] refers to persuasion, inquiry and information-seeking, [151] refers to inquiry and information-seeking, [165] refers to persuasion, information-seeking and deliberation, and [109, 139] refer to inquiry and deliberation). This can be explained taking into account that: (1) information-seeking and inquiry dialogues are similar types, with the difference being mainly that information-seeking dialogues should start with a question [155]; (2) deliberation can be examined both in conflicting (persuasion) and cooperative (inquiry) scenarios between agents, with the difference being that it focuses on deciding about an action that agents should take rather than the validity of a topic of discussion. With regard to negotiation dialogical applications, [16] focused on the persuasive aspect of negotiations and the goal in [111] was primarily the study of human real-life communication rather than the application itself. The number of works and what they concentrate on demonstrates that practical implementations of such dialogues has been under-examined.

Most of the applications that concern dialogue types other than persuasion come with a simple User Interface [33, 46, 109, 111, 121, 200]. In most cases this is because the main focus is either developing or examining a theoretical argumentation framework for dialogues and/or investigating if it is feasible to implement it as an actual application [33, 46, 109, 111, 200], or inspecting specific argumentation software tools [121].

Sections 3 and 4 discussed tools for building argumentation-based reasoners, for example DGEP (discussed in [18, 142, 174]), WFS (discussed in [139]) and Tweety (discussed in [109]), as well as tools used for instantiating agents, for example JADE (discussed in [200]), $Jack^{TM}$ (discussed in [16]) and JaCaMo (discussed in [151]). Concerning the latter, we have also encountered Dial4JaCa (leveraged by [76]), a communication interface integrating JaCaMo and Google Dialogflow[32]. These are more sophisticated software tools compared to the simple use of programming languages for application development purposes (e.g., [33, 46, 111]), and bring elements of agent theory into the implementations. No-

[32] `https://dialogflow.com/`.

tice that only a small number of existing applications have attempted to use NLP (i.e., [21, 81, 142]) or adopt a chatbot-like approach (i.e., [76]) in these dialogical applications.

It is worth observing that, within the surveyed literature, only a handful of argumentation-based dialogue implementations clearly harnessed panoptic engines or solvers as described in Sections 3.2, and 3.3. In particular, two of such research [49, 52] incorporate the ASPARTIX solver ([75] an older version of the latest [74]), whereas a third study [171] structures its main argumentative module (ArgHRI) by embedding the results of ArgTrust v1.0 [180]. Although this does not exclude dialogue systems that merge reasoning engine components with other elements in their overall architecture (which constitutes the majority of our findings), it is still surprising that we did not identify more dialectical applications employing argumentation engines, given the subsisting straightforward connection between the two.

One of the factors included in our analysis methodology is the use of data sets in dialogical applications. It is interesting to see that this component is not taken into account by all the implementations reviewed as we might have expected. Instead, applications such as the ones described in [33, 46, 109] deal with arguments leaving out of the conversation the employment of specific domains. Regarding the data sets visited, [18, 121, 174] use the AIFdb [119] database which deals with the storage and access of AIF argument structures [59], whereas non-argumentative data sets were also found, such as ACKTUS (a web-based tool for modelling medical knowledge into rules and claims in natural language [126, 127]), fraud scenarios from the scenario library and the repository of crime report from the National Service Centre E-Crime Dutch Police (discussed in [21, 142]), and EdiC: the Estonian Dialogue Corpus which comprises of different kinds of human-human dialogues (discussed in [111]).

On the evaluation side, many of the works assessed use formal proofs for appraising their applications as it is common that they implement existing dialogue systems from the literature, or prove different properties for their systems, for example [16, 33, 46, 81, 109, 121, 142, 151, 200]. Note, even in papers where this is not explicitly stated, we assume that this occurs as the dialogue systems employed come with proven features. The use of formal proofs demonstrates the value of the results of theoretical dialogue systems investigated in the argumentation research. However, it is also important to assess the functionality of an application itself, especially when it involves interactive systems such as chatbots. Indeed, their primary goal is direct communication with the user, thus, the most suited evaluation should occur via tests with human participants, as, for example, is done in [6, 54, 55, 96, 111, 163].

Finally, many of the works reviewed either describe the dialogue protocol they follow (e.g., [16, 81, 139, 150, 171, 200]), or this is implicit as the authors refer the dialogue system they leverage (e.g., [33, 46, 111, 121]). The ones that do not refer to a protocol are concentrated on describing software tools (e.g., [18, 21, 174]), or other theoretical properties of the dialogue discussed (e.g., [109, 142]). The characteristics of the moves as well as the dialogue history (also referred to as commitment store) are specified too in the works where the protocol of the dialogue is examined. The component of strategy, however, is not visited that often in applications that concern non-persuasion dialogue types. For example, [33] and [46] refer to winning strategies based on the dialogue games they implement, but both of these papers examine persuasion dialogues. One exception is [200], which examines inquiry dialogues, but provides strategies for avoiding endless dialogues, finishing a dialogue quickly and resolving conflicts. Finally, ,we note that structured argumentation is mainly employed in the applications reviewed (e.g., [16, 46, 49, 52, 121, 139, 142, 151, 165, 200]) in comparison to abstract argumentation (e.g., [33, 81, 109, 171]).

6 Future directions

This section focuses on two key emerging areas for future work in the application of argumentation-based dialogues. The first is the use of enthymemes (Section 6.1), to handle incomplete arguments. The second is the use of argumentation to resolve current issues with LLM implementations (Section 6.2).

6.1 Enthymemes

As mentioned earlier, enthymemes are arguments that lack a complete logical structure. This means that one may omit one or more premises or inference rules, or the claim of the argument they intend to get across to their discussant. This might be because they expect the recipient of the 'incomplete' argument to understand its missing elements based on information they share, or previous conversations they had. Nevertheless, it is not always certain that the recipient of an enthymeme E is able to reconstruct correctly the intended 'complete' argument A from which E was generated. There might be multiple ways to complete E, e.g. the recipient of E might assume that E is part of an intended 'complete' argument B and fill the gaps with parts of B instead of A. Consider for example the following dialogue [197]:

Example 1.

1. Bob: *You can't afford to eat at a restaurant today.*
2. Alice: *Why not?*
3. Bob: *Because you owe money and if you owe money then you probably can't afford to eat at a restaurant.*
4. Alice: *I made a deal with my creditors.*
5. Bob: *So what?*
6. Alice: *So I don't need to pay the bills today.*
7. Bob: *Why is that relevant?*
8. Alice: *I thought that the reason you thought I owe money is because I have bills to pay today. Hence, I can't afford to eat at a restaurant today.*
9. Bob: *No! I meant that you owe money because you need to pay Kate back today. So, you can't afford to eat at a restaurant today.*

Bob first asserts a claim without any supporting premises (1). The reasons for believing the claim are not clear to Alice, so she asks for clarification (2), which Bob provides (3). Notice that, when combined, (1) and (3) form a complete argument, hence they can both be considered enthymemes for this complete argument. Alice then presents an enthymeme (4) for an argument that she believes counters the argument Bob is making. Note that the enthymeme Alice presents does not explicitly contradict anything that Bob has said, and so Bob asks for clarification (5) on what she is meant to infer from this enthymeme, which Alice provides (6). However, Alice's clarification still does not explicitly contradict anything Bob has said. Since Bob does not understand why Alice's enthymeme is relevant to what he said, he asks Alice to explain what she thought he meant (7). Alice explains the assumption she had made (8), which Bob then corrects (9).

This simple example illustrates the need for a dialogue system that allows human and/or computational agents to both 'backward extend' enthymemes (where missing premises are provided in 3 above) and 'forward extend' enthymemes (where missing inferences are given, as in 6), and to request such extensions (2 and 5). It also warrants the need for allowing agents to ask what another agent has assumed was intended by an enthymeme (7), to answer such a question (8), and to correct any erroneous assumptions (9).

Work on how enthymemes are handled during dialogues between human and/or computational agents is another area that is not heavily studied. Notable exceptions include the work of Black and Hunter [27], De Saint-Cyr [68], Hosseini [100], Xydis et al. [197, 198, 199], Odekerken et al. [141] and Leiva et al. [123]. From these works, [27, 100, 123, 141] employ locutions that capture only the backward extension of enthymemes, [198] makes use of locutions used to handle only the forward extension of enthymemes, and [199] focuses on capturing the misunderstandings that may occur during the dialogue, whereas [141] does not specify locutions, but explores the notion of "queryable literals" which essentially enable dealing with backward extension. Only [68] and [197] address both backward and forward extension of enthymemes, whereas [197] additionally enables resolution of misunderstandings that arise due to use of enthymemes.

Note that Prakken's dialogue system for persuasion discussed in [157] (described previously in Section 2.2.2) can also be perceived as a dialogue system which accounts for enthymemes since it includes locutions which support backward extension of enthymemes (e.g. *why* and *since*), as does the work discussed in [136]. Both of these works, also, refer to how the outcome of the dialogue relates to the AF that is instantiated based on contents of the enthymemes moved during a dialogue, with the former providing soundness and completeness results and the latter making a conjecture that such results hold for their system. Likewise, the authors in [198, 199] show soundness and completeness results for their respective systems. This is important as it confirms that there is no disadvantage to the use of enthymemes in dialogue and ensures that the dialogue can be played out such that an enthymeme moved in the dialogue is only justified in the case that its intended argument is justified by the contents of the moves made in the dialogue.

Not many practical applications on argumentation-based dialogues account for the use of enthymemes. We believe that more applications implementing argumentation-based dialogues that allow the handling of enthymemes should be developed. Although enthymemes' ubiquity poses a significant challenge when it comes to applying them in formal dialogues and verifying their acceptability status during these dialogues (e.g., in [141] it is explained how querying –or else requesting a backward extension for– all possible premises can be computationally challenging, however a sound approximation alternative is presented), humans are able to manage the use of enthymemes in their everyday life and assess them correctly during their communication (as displayed in Example 1). Therefore, if we are to develop computational dialogue systems and applications implementing them which reflect people's dialogical interactions and produce accurate results on the evaluation of their utterances, we need to formally incorporate enthymemes in sound and complete dialogues.

In [171], a persuasion dialogue protocol is presented where a participating agent Ag_1 commences a dialogue by *asserting* an argument $A = (S, c)$, where S is the set of premises of A and c is the conclusion of A. Then, their interlocutor Ag_2 can either *accept*, or *challenge*, or *attack* A. In case Ag_2 *challenges* A, it means that Ag_2 requests a supporting argument for either a premise $p \in S$ of A or the claim c of A. Ag_1 can fulfill the request of Ag_2 by *asserting* (i.e., providing) an argument B that either supports p (i.e., $B = (S', p)$) or c (i.e., $B = (S', c)$) depending on the request of Ag_2. If B supports p, it is easy to see that by combining arguments A and B, an argument $C = (S' \cup S, c)$ can be instantiated.

Although the authors in [171] assume that their arguments are complete, the locutions employed in the aforementioned scenario can be used to model backward extension, both requesting (with a *challenge* move) and providing it (with an *assert* move). Specifically, we can consider A and B as enthymemes of the intended complete argument C. In other words, [171] already captures an instance of enthymeme handling in argumentation-based dialogues. In [197], the authors present a dialogue protocol which accounts for both backward and forward extension of enthymemes, as well as resolution of misunderstandings that may occur between the participants of a dialogue. We believe that by expanding the set of locutions in [171] and the persuasion dialogue protocol introduced in that work, it will be possible to additionally capture and deal with both forward extension of enthymemes and misunderstandings that may occur during the dialogue due to the use of enthymemes. The latter is of particular importance, as in case that a misunderstanding has already taken place, the participants can backtrack to that point of the dialogue, resolve the misunderstanding and still reach the "correct" conclusions/decisions based on the knowledge they have shared[33]. Modifying the locutions and the protocol introduced in [171] are two lines of research which the authors of this chapter are actively exploring as part of our ongoing work.

6.2 Improving the performance of large-language models

The recent significant increase in popularity of Artificial Intelligence is largely due to the surge of *Large Language Models (LLMs)* and their outstanding performance against multiple benchmarks. Essentially, a *language model (LM)* is primarily designed to predict tokens based on the likelihood of their occurrences given previous word sequences. Stemming from statistical learning methods and recurrent neural networks, it was eventually the Transformer architecture [188] that consolidated the paradigm shift of 'pre-training' and 'fine-tuning' a language model on large datasets, ultimately leading to the development of LLMs [201]. Indeed, researchers discovered how scaling the internal structure or the

[33]See [199] for a system that focuses on dealing with misunderstandings, and discusses soundness and completeness results in persuasion dialogues. Such results concern the acceptability of arguments and enthymemes moved in the dialogue and the argument framework instantiated by the contents of the moves, under some semantics σ.

training data size results in enhanced capabilities compared to smaller versions of the same model [60, 99, 110]. For example, LLMs prove to outperform most of the previous standards and predecessors within the scope of information extraction [124], natural language inference, question answering, dialogue tasks [161] and machine translation [108].

A noteworthy instance of this new technology is the well-known ChatGPT[34], which hinges on the GPT model family [35, 144, 145], although many other LLMs are regarded as having similar performance levels [7, 89, 135]. The trade-off for such impressive accomplishments consists of multiple shortcomings that likewise affect each large language model. Among these weaknesses, we highlight: hallucinations [107], emergent abilities [194], biased and toxic output (along with the challenging task of models-humans values alignment) [35], lack of transparency in response generation, high cost of training and carbon footprint emissions. In addition, every LLM is limited in its knowledge of the world to its pre-training data, thus leaving a gap concerning up-to-date information that can only be covered by resorting to external tools or plugins (usually involving web search or retrieval-augmented-generation, RAG, capabilities [87]). Furthermore, it has also been shown how models such as GPT-3 fall short of producing adequate and compelling arguments [98]. The authors of such a study elaborate this conclusion after a thorough application of the Comprehensive Assessment Procedure for Natural Argumentation (CAPNA) protocol [97]. GPT-3 is able to produce different argument types (thus identifying common human dialectical patterns), but it fails when it comes to providing their acceptability, mostly generating fallacious arguments. The entailed consequence is that the capability of arguing, intended as an exchange of reasoning between intelligent entities, should be learnt by AIs if their purpose aims for more than just acquiring and repeating information.

In the following, drawing from the insights outlined in [50], we show how the employment of computational argumentation-based dialogical approaches may result in promising solutions for issues in current LLMs. Aside from an overall improvement in the quality of the posited arguments, LLMs can achieve different benefits from combining with computational argumentation [53].

[34] `https://chat.openai.com/`.

Transparency in response generation. Given the current 'black-box' nature of LLMs and the complexity of understanding their output generation (especially for laypeople), there is a present urge to provide clear explanations about what drives AIs' decisions. The goal of overcoming this lack of transparency is among the reasons that foster research within the thriving field of eXplaninable AI (XAI), where argumentative strategies are proposed as adequate forms of justifications [65, 187]. These intuitions are backed by studies such as [47, 51, 132] where it is suggested that AI systems should adopt an argumentation-based approach to explanations consisting of dialogue protocols characterising the interactions between an explainer and an explainee. Embedded in LLMs, such a dialectical interplay would provide an informative post hoc method to deliver deliberated explanations to end-users while also ensuring detailed replies to follow-on queries. Contrary to the study of Turpin et al. [186], we believe such a formal argumentative approach to be capable of producing and rationalising unbiased explanations by filtering, following Dung's semantics [71], the unacceptable ones.

Notice that even the renowned GPT-4 exhibits drawbacks when dealing with the *process consistency* of its explanations: it provides a plausible account of the rationale behind the generation of its output, but it often fails in representing a more general justification able to predict the outcome of the model given similar inputs [144]. An argumentative dialogue (such as the Explanation-Question-Response, EQR, protocol [49, 51], previously mentioned in Section 4.2) designed for explanation purposes would allow solving the process-consistency issues by providing conversations where more information can be retrieved and thus eschewing the limited explanation length and language constraints deemed to be the leading causes of the problem [38].

Hallucination. Defined as *"generated content that is nonsensical or unfaithful to the provided source content"* [107], the phenomenon of hallucination in natural language generation can be divided into *intrinsic* and *extrinsic*. The former refers to generated output that contradicts the source upon which the LLM was trained. The second, instead, represents an output that cannot be verified. The employment of argumentative XAI dialogical methods can assist in probing the model replies, thus, potentially identifying and filtering out hallucinating contents, or granting, in the worst-case scenario, the retrieval of additional information over the produced content.

Emergent abilities. The occurrence of these unpredictable phenomena consists of the unexpected appearance of specific competencies in large-scale models that do not manifest in smaller ones. Thus, it is not possible to anticipate the "emergence" of these abilities[35] (e.g., improved arithmetic, multi-task understanding, enhanced multilingual operations) by simply examining smaller-scale models [194]. Leveraging argumentative XAI dialogical methods (e.g. the aforementioned EQR protocol [51]) could indirectly help as a post hoc solution: although it cannot identify the reasons why emergent abilities originate, it could nonetheless provide explanations that would clarify their functioning. Notice that, although inexplicable, emergent abilities usually characterise useful competencies acquired by a model, in contrast with hallucinations that only refer to contradictory or made-up textual facts provided by the LLM as a reply to a user prompt.

7 Conclusion

This chapter set out to review applications of argumentation-based dialogue, and took a broad view as to what this meant. Viewing "dialogue" as meaning "an exchange of ideas and opinions"[36], we see it as covering any such exchange between two or more humans or agents (in any combination) or even the internal reasoning process of a single human or agent (though we do not focus on the latter). To make this chapter relatively self-contained, we briefly covered (Section 2) the elements of formal argumentation and dialogue games that we felt were required to understand the rest of the paper, before beginning the review proper by discussing (Section 3) components of argumentation-based dialogue systems such as solvers of various kinds and chatbots.

[35] Emergent abilities constitute a controversial topic and some studies even argue against their existence [166].

[36] Meaning 2b in Merriam Webster `https://www.merriam-webster.com/dictionary/dialogue` at the time of writing.

Section 4 then contains the main body of the review, looking at current work on applications of argumentation-based dialogue. It starts (see Section 4.1) by providing a description of the way that we went about the analysis of the systems that we found in the literature. Next (Section 4.2), we look at one specific application, indulging ourselves by taking this from our work on the CONSULT project, which we think nicely illustrates many of the features of a typical use of argumentation-based dialogue. Following that, we look (Section 4.3) at a number of applications that are built on top of work on formal dialogue models, many of them fitting neatly into the typology introdcued by Walton and Krabbe. As we argue, these are systems that fit the more traditional approach in the computational argumentation community. Then, finally (Section 4.4), we consider chatbots that are based around the use of argumentation. These we consider to be a more recent development, following the growth in ML-based chatbots.

Section 5 then discusses key themes that cut across all this work, and Section 6 digs into the detail of two areas of future work — enthymemes and the benefit of combining large language models with argumentation-based dialogues. These two areas are ones we find particularly exciting, and plan to pursue work in them ourselves.

Acknowledgements

This work was partially supported by the UK Engineering & Physical Sciences Research Council (EPSRC) under grant #EP/P010105/1, and by a grant from the University of Lincoln.

References

[1] Eleni Adamopoulou and Lefteris Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020.

[2] Mario Alviano. The PYGLAF argumentation reasoner (ICCMA2021), 2021. `http://argumentationcompetition.org/2021/downloads/pyglaf.pdf`.

[3] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, pages 197–215, 2002.

[4] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, pages 1062–1093, 2008.

[5] Leila Amgoud, Simon Parsons, and Nicolas Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 338–342, 2000.

[6] Pierre Andrews, Suresh Manandhar, and Marco De Boni. Argumentative human computer dialogue for automated persuasion. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 138–147, 2008.

[7] Anthropic. Claude 3.5. *Anthropic Blog*, 2024. `https://www.anthropic.com/news/claude-3-5-sonnet` (last accessed 24/07/2024).

[8] AOS. Agent manual, release 5.3. http://https://aosgrp.com.au/jack/, June 2005.

[9] Gilles Audemard and Laurent Simon. On the glucose SAT solver. *International Journal on Artificial Intelligence Tools*, 27(01):1840001, 2018.

[10] M Q Azhar and Elizabeth I Sklar. A study measuring the impact of shared decision making in a human-robot team. *International Journal of Robotics Research (IJRR)*, 36:461–482, 2017.

[11] K Bala, Mukesh Kumar, Sayali Hulawale, and Sahil Pandita. Chat-bot for college management system using ai. *International Research Journal of Engineering and Technology*, 4(11):2030–2033, 2017.

[12] Panos Balatsoukas, Talya Porat, Isabel Sassoon, Kai Essers, Nadin Kokciyan, Martin Chapman, Archie Drake, Sanjay Modgil, Mark Ashworth, Elizabeth Sklar, Simon Parsons, and Vasa Curcin. User involvement in the design of a data-driven self-management decision support tool for stroke survivors. In *18th IEEE International Conference on Smart Technologies*, pages 1–6, 2019.

[13] Pietro Baroni, Dov Gabbay, Massimilino Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*, volume 1. College Publications, London, England, 2018.

[14] Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, pages 429–448, 2003.

[15] Trevor JM Bench-Capon and Paul E Dunne. Argumentation in artificial intelligence. *Artificial intelligence*, pages 619–641, 2007.

[16] Jamal Bentahar and Jihad Labban. An argumentation-driven model for flexible and efficient persuasive negotiation. *Group Decision and Negotiation*, 20:411–435, 2011.

[17] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.

[18] Floris Bex, John Lawrence, and Chris Reed. Generalising argument dialogue with the dialogue game execution platform. In *Proceedings of the Conference on Computational Models of Argument*, pages 141–152, 2014.

[19] Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. Implementing the argument web. *Communications of the ACM*, pages 66–73, 2013.

[20] Floris Bex, Sanjay Modgil, Henry Prakken, and Chris Reed. On logical reifications of the argument interchange. *Journal of Logic and Computation*, 2013.

[21] Floris Bex, Joeri Peters, and Bas Testerink. AI for online criminal complaints: From natural dialogues to structured scenarios. In *Workshop on Artificial intelligence for Justice (ECAI 2016)*, page 22, 2016.

[22] Floris Bex and Chris Reed. Dialogue templates for automatic argument processing. In *Proceedings of the Conference on Computational Models of Argument*, pages 366–377, 2012.

[23] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of Satisfiability*, volume 185. IOS press, 2009.

[24] Stefano Bistarelli, Lars Kotthoff, Francesco Santini, and Carlo Taticchi. Summary report for the Third International Competition on Computational Models of Argumentation. In *AI Magazine*, volume 42, pages 70–73, 2021.

[25] Stefano Bistarelli, Fabio Rossi, Francesco Santini, and Taticchi Carlo. CONARG: A constraint-programming solver for abstract argumentation problems, 2021. `http://argumentationcompetition.org/2021/downloads/conarg.pdf`.

[26] Elizabeth Black, Amanda J Coles, and Christopher Hampson. Planning for persuasion. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 933–942, 2017.

[27] Elizabeth Black and Anthony Hunter. Using enthymemes in an inquiry dialogue system. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-agent Systems*, pages 437–444, 2008.

[28] Elizabeth Black and Anthony Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, pages 173–209, 2009.

[29] Elizabeth Black and Anthony Hunter. A relevance-theoretic framework for constructing and deconstructing enthymemes. *Logic and Computation*, pages 55–78, 2012.

[30] Elizabeth Black, Nicolas Maudet, and Simon Parsons. Argumentation-based dialogue. In Dov Gabbay, Massimiliano Giacomin, Guillermo R.

Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, page 511. College Publications, 2021.

[31] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, pages 747–761, 2013.

[32] Andrei Bondarenko, Phan Minh Dung, Robert A Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, pages 63–101, 1997.

[33] Richard Booth, Martin Caminada, and Braden Marshall. Disco: A web-based implementation of discussion games for grounded and preferred semantics. In *Proceedings of the Conference on Computational Models of Argument*, pages 453–454. 2018.

[34] AnneMarie Borg and Daphne Odekerken. PyArg for solving and explaining argumentation in python: Demonstration. In *Proceedings of the Conference on Computational Models of Argument*, pages 349–350. 2002.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[36] Daniel Bryant and Paul Krause. A review of current defeasible reasoning implementations. *The Knowledge Engineering Review*, 23(3):227–260, 2008.

[37] Daniel Bryant, Paul J Krause, and Gerard Vreeswijk. Argue tuProlog: A lightweight argumentation engine for agent applications. In *Proceedings of the Conference on Computational Models of Argument*, pages 27–32. 2006.

[38] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

[39] Guendalina Caldarini, Sardar Jaf, and Kenneth McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1):41, 2022.

[40] Martin Caminada. On the issue of reinstatement in argumentation. In *European Workshop on Logics in Artificial Intelligence*, pages 111–123, 2006.

[41] Martin Caminada. A discussion game for grounded semantics. In *Proceedings of the 3rd International Workshop on Theory and Applications of Formal Argumentation*, pages 59–73, 2015.

[42] Martin Caminada. Argumentation semantics as formal discussion. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, volume 1, pages 487—-518. College Publications, 2018.

[43] Martin Caminada, Wolfgang Dvořák, and Srdjan Vesic. Preferred semantics as socratic discussion. *Journal of Logic and Computation*, pages 1257–1292, 2014.

[44] Martin Caminada, Sanjay Modgil, and Nir Oren. Preferences and unrestricted rebut. In *Proceedings of the Conference on Computational Models of Argument*, pages 209–220, 2014.

[45] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22:64–102, 2011.

[46] Martin Caminada and Sören Uebis. An implementation of argument-based discussion using ASPIC. In *Proceedings of the Conference on Computational Models of Argument*, pages 455–456. 2020.

[47] Federico Castagna. Towards a fully-fledged formal protocol for the Explanation-Question-Response dialogue. In *Online Handbook of Argumentation for AI*, pages 17–21, 2022.

[48] Federico Castagna. Dialectical argument game proof theories for classical logic. *Journal of Applied Logics*, 2631(3):279, 2023.

[49] Federico Castagna, Alexandra Garton, Peter Mcburney, Simon Parsons, Isabel Sassoon, and Elizabeth I Sklar. EQRbot: a chatbot delivering EQR argument-based explanations. In *Frontiers in Artificial Intelligence*, volume 6. Frontiers, 2023.

[50] Federico Castagna, Nadin Kokciyan, Isabel Sassoon, Simon Parsons, and Elizabeth Sklar. Computational Argumentation-based Chatbots: a Survey. *Journal of Artificial Intelligence Research*, 80, 2024.

[51] Federico Castagna, Peter McBurney, and Simon Parsons. Explanation–Question–Response dialogue: An argumentative tool for explainable AI. *Argument & Computation*, (in press).

[52] Federico Castagna, Simon Parsons, Isabel Sassoon, and Elizabeth I. Sklar. Providing explanations via the EQR argument scheme. In *Proceedings of the Conference on Computational Models of Argument*, 2022.

[53] Federico Castagna, Isabel Sassoon, and Simon Parsons. Can formal argumentative reasoning enhance LLM's performances? https://arxiv.org/abs/2405.13036, 2024.

[54] Lisa Andreevna Chalaguine and Anthony Hunter. A persuasive chatbot using a crowd-sourced argument graph and concerns. In *Proceedings of*

the Conference on Computational Models of Argument*, pages 9–20, 2020.

[55] Lisa Andreevna Chalaguine and Anthony Hunter. Addressing popular concerns regarding COVID-19 vaccination with natural language argumentation dialogues. In *European Conference on Symbolic and Quantitative Approaches with Uncertainty*, pages 59–73, 2021.

[56] Lisa Andreevna Chalaguine, Anthony Hunter, Henry Potts, and Fiona Hamilton. Impact of argument type and concerns in argumentation with a chatbot. In *Proceedings of the 31st IEEE International Conference on Tools with Artificial Intelligence*, pages 1557–1562, 2019.

[57] Martin Chapman, G Abigail, Isabel Sassoon, Nadin Kökciyan, Elizabeth I Sklar, Vasa Curcin, et al. Using microservices to design patient-facing research software. In *Proceedings of the 18th IEEE International Conference on e-Science*, pages 44–54, 2022.

[58] Martin Chapman, Panagiotis Balatsoukas, Nadin Kökciyan, Kai Essers, Isabel Sassoon, Mark Ashworth, Vasa Curcin, Sanjay Modgil, Simon Parsons, and Elizabeth I Sklar. Computational argumentation-based clinical decision support. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 2345–2347, 2019.

[59] Carlos Chesnevar, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, Steven Willmott, et al. Towards an argument interchange format. *The Knowledge Engineering Review*, pages 293–316, 2006.

[60] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[61] Codecademy. What are chatbots? 2022. `https://www.codecademy.com/article/what-are-chatbots` (last accessed 11/10/2023).

[62] Eva Cogan, Simon Parsons, and Peter McBurney. What kind of argument are we going to have today? In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems*, pages 544–551, 2005.

[63] Dennis Craandijk and Floris Bex. Deep learning for abstract argumentation semantics. *arXiv preprint arXiv:2007.07629*, 2020.

[64] Kristijonas Cyras, Brendan Delaney, Denys Prociuk, Francesca Toni, Martin Chapman, Jesús Domínguez, and Vasa Curcin. Argumentation for explainable reasoning with conflicting medical recommendations. In *Proceedings of the CEUR Workshop*, volume 2237. CEUR-WS, 2018.

[65] Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and

Francesca Toni. Argumentative XAI: a survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Survey Track*, pages 4392–4399, 2021.

[66] Raimund Dachselt, Sarah Alice Gaggl, Markus Krötzsch, Julian Mendez, Dominik Rusovac, and Mei Yang. Nexas: A visual tool for navigating and exploring argumentation solution spaces. In *Proceedings of the Conference on Computational Models of Argument*, pages 116–127, 2022.

[67] Richard Dawkins. *The Selfish Gene.* Oxford University Press, 40th anniversary edition, 2016.

[68] Florence Dupin de Saint-Cyr. Handling enthymemes in time-limited persuasion dialogs. In *Proceedings of the International Conference on Scalable Uncertainty Management*, pages 149–162, 2011.

[69] Jürgen Dix, Simon Parsons, Henry Prakken, and Guillermo Ricardo Simari. Research challenges for argumentation. *Computer Science Research and Development*, 23(1):27–34, 2009.

[70] Didier Dubois and Henri Prade. Possibilistic logic: a retrospective and prospective view. *Fuzzy sets and Systems*, pages 3–23, 2004.

[71] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, pages 321–357, 1995.

[72] Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 199–218. 2009.

[73] Phan Minh Dung and Phan Minh Thang. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proceedings of the Conference on Computational Models of Argument*, pages 171–182, 2010.

[74] Wolfgang Dvorák, Matthias König, Johannes P. Wallner, and Stefan Woltran. ASPARTIX-V21, 2021. `http://argumentationcompetition.org/2021/downloads/aspartix-v21.pdf`.

[75] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. ASPARTIX: Implementing argumentation frameworks using answer-set programming. In *International Conference on Logic Programming*, pages 734–738. Springer, 2008.

[76] Débora C Engelmann, Alison R Panisson, Renata Vieira, Jomi Fred Hübner, Viviana Mascardi, and Rafael H Bordini. MAIDS — a framework for the development of multi-agent intentional dialogue systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1209–1217, 2023.

[77] Kai Essers, Martin Chapman, Nadin Kokciyan, Isabel Sassoon, Talya Po-

rat, Panagiotis Balatsoukas, Peter Young, Mark Ashworth, Vasa Curcin, Sanjay Modgil, et al. The CONSULT system. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, pages 385–386, 2018.

[78] Marcelo Alejandro Falappa, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. Belief revision and argumentation theory. In I Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 341–360. 2009.

[79] Xiuyi Fan and Francesca Toni. Agent strategies for ABA-based information-seeking and inquiry dialogues. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 324–329, 2012.

[80] Xiuyi Fan and Francesca Toni. A general framework for sound assumption-based argumentation dialogues. *Artificial Intelligence*, pages 20–54, 2014.

[81] Bettina Fazzinga, Andrea Galassi, and Paolo Torroni. A privacy-preserving dialogue system based on argumentation. *Intelligent Systems with Applications*, 16:200113, 2022.

[82] Johannes K. Fichte, Markus Hecher, Piotr Gorczyca, and Ridhwan Dewoprabowo. A-FOLIO DPDB – system description for ICCMA 2021, 2021. `http://argumentationcompetition.org/2021/downloads/a-folio-dpdb.pdf`.

[83] Johannes K Fichte, Markus Hecher, Patrick Thier, and Stefan Woltran. Exploiting database management systems and treewidth for counting. *Theory and Practice of Logic Programming*, 22(1):128–157, 2022.

[84] John Fox, David Glasspool, Dan Grecu, Sanjay Modgil, Matthew South, and Vivek Patkar. Argumentation-based inference and decision making–a medical perspective. *IEEE Intelligent Systems*, 22(6):34–41, 2007.

[85] Dorian Gaertner and Francesca Toni. CaSAPI: a system for credulous and sceptical argumentation. In *Proceedings of the Workshop on Non-monotonic Reasoning*, pages 80–95, 2007.

[86] Sarah A Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. *Artificial Intelligence*, 279:103193, 2020.

[87] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[88] Alejandro J García and Guillermo R Simari. Defeasible logic program-

ming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[89] Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. https://arxiv.org/abs/2403.05530, 2024.

[90] Esteban Guerrero, Juan Carlos Nieves, and Helena Lindgren. An activity-centric argumentation framework for assistive technology aimed at improving health. *Argument & Computation*, pages 5–33, 2016.

[91] Kai Guo, Jian Wang, and Samuel Kai Wah Chu. Using chatbots to scaffold EFL students' argumentative writing. *Assessing Writing*, 54:100666, 2022.

[92] Christos Hadjinikolis, Yiannis Siantos, Sanjay Modgil, Elizabeth Black, and Peter McBurney. Opponent modelling in persuasion dialogues. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 164–170, 2013.

[93] Emmanuel Hadoux and Anthony Hunter. Strategic sequences of arguments for persuasion using decision trees. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1128–1134, 2017.

[94] Emmanuel Hadoux and Anthony Hunter. Comfort or safety? gathering and using the concerns of a participant for better persuasion. *Argument & Computation*, 10(2):113–147, 2019.

[95] Emmanuel Hadoux, Anthony Hunter, and Sylwia Polberg. Strategic argumentation dialogues for persuasion: Framework and experiments based on modelling the beliefs and concerns of the persuadee. *arXiv preprint arXiv:2101.11870*, 2021.

[96] Christian Hauptmann, Adrian Krenzer, Justin Völkel, and Frank Puppe. Argumentation effect of a chatbot for ethical discussions about autonomous AI scenarios. *Knowledge and Information Systems*, 66:3607–3637, 2024.

[97] Martin Hinton. CAPNA—the comprehensive assessment procedure for natural argumentation. In *Evaluating the Language of Argument*, volume 37 of *Argumentation Library*, pages 167–194. 2021.

[98] Martin Hinton and Jean HM Wagemans. How persuasive is AI-generated argumentation? an analysis of the quality of an argumentative text produced by the GPT-3 AI text generator. *Argument & Computation*, 14(1):59–74, 2023.

[99] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Au-

relia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. https://arxiv.org/pdf/2203.15556, 2022.

[100] Seyed Ali Hosseini. *Dialogues incorporating enthymemes and modelling of other agents' beliefs*. PhD thesis, King's College London, 2017.

[101] Nguyen Duy Hung. Inference procedures and engine for probabilistic argumentation. *International Journal of Approximate Reasoning*, 90:163–191, 2017.

[102] Anthony Hunter. Real arguments are approximate arguments. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 66–71, 2007.

[103] Anthony Hunter. Towards a framework for computational persuasion with applications in behaviour change. *Argument & Computation*, 9(1):15–40, 2018.

[104] Anthony Hunter, Lisa Chalaguine, Tomasz Czernuszenko, Emmanuel Hadoux, and Sylwia Polberg. Towards computational persuasion via natural language argumentation dialogues. In *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 18–33. Springer, 2019.

[105] Anthony Hunter and Nico Potyka. Updating probabilistic epistemic states in persuasion dialogues. In *Proceedings of the 14th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 46–56, 2017.

[106] Matti Järvisalo, Tuomo Lehtonen, and Andreas Niskanen. Design of ICCMA 2023, 5th International Competition on Computational Models of Argumentation: A preliminary report. In *Proceedings of the First lnternational Workshop on Argumentation and Applications*, pages 4–10, 2023.

[107] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248, 2023.

[108] Wenxiang Jiao, Wenxuan Wang, JT Huang, Xing Wang, and ZP Tu. Is ChatGPT a good translator? yes with GPT-4 as the engine. https://arxiv.org/pdf/2301.08745, 2023.

[109] Timotheus Kampik and Dov Gabbay. Towards DiArg: An argumentation-based dialogue reasoning engine. In *Proceedings of the Third International Workshop on Systems and Algorithms for Formal Argumentation*, pages 14–21, 2020.

[110] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. https://arxiv.org/pdf/2001.08361, 2020.

[111] Mare Koit and Haldur Õim. A computational model of argumentation in agreement negotiation processes. *Argument & Computation*, 6(2):101–129, 2015.

[112] Eric M Kok, John-Jules Ch Meyer, Henry Prakken, and Gerard AW Vreeswijk. A formal argumentation framework for deliberation dialogues. In *Proceedings of the 7th International Workshop on Argumentation in Multi-Agent Systems*, pages 31–48, 2011.

[113] Nadin Kökciyan, Martin Chapman, Panagiotis Balatsoukas, Isabel Sassoon, Kai Essers, Mark Ashworth, Vasa Curcin, Sanjay Modgil, Simon Parsons, and Elizabeth I Sklar. A collaborative decision support tool for managing chronic conditions. In *Proceedings of the 17th World Congress on Medical and Health Informatics*, pages 644–648, 2019.

[114] Nadin Kökciyan, Simon Parsons, Isabel Sassoon, Elizabeth Sklar, and Sanjay Modgil. An argumentation-based approach to generate domain-specific explanations. In *Proceedings of the European Conference on Multi-Agent Systems*, pages 319–337. Springer, 2020.

[115] Nadin Kökciyan, Isabel Sassoon, Elizabeth Sklar, Sanjay Modgil, and Simon Parsons. Applying metalevel argumentation frameworks to support medical decision making. *IEEE Intelligent Systems*, 36(2):64–71, 2021.

[116] Jean-Marie Lagniez, Emmanuel Lonca, and Jean-Guy Mailly. CoQuiAAS: A constraint-based quick abstract argumentation solver. In *Proceedings of the 27th IEEE International Conference on Tools with Artificial Intelligence*, pages 928–935, 2015.

[117] Jean-Marie Lagniez, Emmanuel Lonca, and Jean-Guy Mailly. Crustabri, the evolution of CoQuiAAS. In *Solver and Benchmark Descriptions of ICCMA 2023 : 5th International Competition on Computational Models of Argumentation*, page 20. 2023. http://hdl.handle.net/10138/565357.

[118] Jean-Marie Lagniez, Emmanuel Lonca, Jean-Guy Mailly, and Julien Rossit. Design and results of ICCMA 2021. https://arxiv.org/pdf/2109.08884, 2021.

[119] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. Aifdb: Infrastructure for the argument web. In *Proceedings of the Conference on Computational Models of Argument*, pages 515–516, 2012.

[120] John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2020.

[121] John Lawrence, Jacky Visser, and Chris Reed. Polemicist: A dialogical interface for exploring complex debates. In *9th International Conference on Computational Models of Argument, COMMA 2022*, pages 365–366, 2022.

[122] Tuomo Lehtonen, Johannes P Wallner, and Matti Järvisalo. ASPforABA–ASP-based algorithms for reasoning in ABA. In *Solver and Benchmark Descriptions of ICCMA 2023 : 5th International Competition on Computational Models of Argumentation*, page 18. 2023. http://hdl.handle.net/10138/565357.

[123] Diego S Orbe Leiva, Sebastian Gottifredi, and Alejandro J García. Automatic knowledge generation for a persuasion dialogue system with enthymemes. *International Journal of Approximate Reasoning*, 160:108963, 2023.

[124] Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. Evaluating ChatGPT's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. https://arxiv.org/abs/2304.11633, 2023.

[125] Helena Lindgren, Timotheus Kampik, Esteban Guerrero Rosero, Madeleine Blusi, and Juan Carlos Nieves. Argumentation-based health information systems: A design methodology. *IEEE Intelligent Systems*, 36(2):72–80, 2020.

[126] Helena Lindgren and Peter Winnberg. A model for interaction design of personalised knowledge systems in the health domain. In *Proceedings of the 3rd International Conference on Electronic Healthcare*, pages 235–242, 2012.

[127] Helena Lindgren and Chunli Yan. Acktus: A platform for developing personalized support systems in the health domain. In *Proceedings of the 5th International Conference on Digital Health 2015*, pages 135–142, 2015.

[128] Michael Luck, Peter McBurney, Onn Shehory, and Steve Willmott. Agent technology, Computing as interaction: A roadmap for agent based computing. AgentLink III, 2005.

[129] Lars Malmqvist. AFGCN v2: A GCN-based approximate solver. In *Solver and Benchmark Descriptions of ICCMA 2023 : 5th International Competition on Computational Models of Argumentation*, page 8. 2023. http://hdl.handle.net/10138/565357.

[130] Peter McBurney, David Hitchcock, and Simon Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22(1):95–132, 2007.

[131] Peter McBurney and Simon Parsons. Dialogue games for agent argu-

mentation. In I Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 261–280. 2009.

[132] Peter McBurney and Simon Parsons. Argument schemes and dialogue protocols: Doug Walton's legacy in Artificial Intelligence. *Journal of Applied Logics*, 8(1):263–286, 2021.

[133] John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial intelligence*, 13(1-2):27–39, 1980.

[134] Hugo Mercier and Dan Sperber. Why do humans reason? Arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34(2):57–74, 2011.

[135] Meta. Introducing Llama 3.1: Our most capable models to date. Meta Blog, 2024. https://ai.meta.com/research/publications/the-llama-3-herd-of-models/.

[136] Sanjay Modgil. Towards a general framework for dialogues that accommodate reasoning about preferences. In *Proceedings of the International Workshop on Theories and Applications of Formal Argumentation*, pages 175–191, 2017.

[137] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. 2009.

[138] Sanjay Modgil and Henry Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[139] Juan Carlos Nieves and Helena Lindgren. Deliberative argumentation for service provision in smart environments. In *Proceedings of the 12th European Conference on Multi-Agent Systems*, pages 388–397, 2015.

[140] Andreas Niskanen and Matti Järvisalo. μ-TOKSIA in ICCMA 2023. In *Solver and Benchmark Descriptions of ICCMA 2023 : 5th International Competition on Computational Models of Argumentation*, page 31. 2023. http://hdl.handle.net/10138/56535.

[141] Daphne Odekerken, Floris Bex, AnneMarie Borg, and Bas Testerink. Approximating stability for applied argument-based inquiry. *Intelligent Systems with Applications*, 16:200110, 2022.

[142] Daphne Odekerken, AnneMarie Borg, and Floris Bex. Estimating stability for efficient argument-based inquiry. In *Proceedings of the Conference on Computational Models of Argument*, pages 307–318, 2020.

[143] Tiago Oliveira, Jérémie Dauphin, Ken Satoh, Shusaku Tsumoto, and Paulo Novais. Argumentation with goals for clinical decision support in multimorbidity. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2031–2033, 2018.

[144] OpenAI. GPT-4 Technical Report. https://arxiv.org/abs/2303.08774, 2024.

[145] OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[146] Oracle. What is a chatbot? https://www.oracle.com/chatbots/what-is-a-chatbot/.

[147] Nir Oren and Timothy J Norman. Arguing using opponent models. In *Proceedings of the International Workshop on Argumentation in Multi-Agent Systems*, pages 160–174, 2009.

[148] Alison R Panisson and Rafael H Bordini. Argumentation schemes in multi-agent systems: A social perspective. In *Proceedings of the 5th International Workshop on Engineering Multi-Agent Systems*, pages 92–108, 2017.

[149] Alison R Panisson, Peter McBurney, and Rafael H Bordini. A computational model of argumentation schemes for multi-agent systems. *Argument & Computation*, 12(3):357–395, 2021.

[150] Alison R Panisson, Peter McBurney, and Rafael H Bordini. Towards an enthymeme-based communication framework. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1708–1710, 2022.

[151] Alison R Panisson, Peter McBurney, and Rafael H Bordini. Towards an enthymeme-based communication framework in multi-agent systems. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 267–277, 2022.

[152] Alison R Panisson, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. An approach for argumentation-based reasoning using defeasible logic in multi-agent programming languages. In *Proceedings of the 11th International Workshop on Argumentation in Multiagent Systems*, pages 1–15, 2014.

[153] S. Parsons and E. I. Sklar. How agents revise their beliefs after an argumentation-based dialogue. In *Proceedings of the 2nd International Workshop on Argumentation in Multiagent Systems*, 2005.

[154] Simon Parsons, Carles Sierra, and Nick Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[155] Simon Parsons, Michael Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[156] Mikolaj Podlaszewski, Martin Caminada, and Gabriella Pigozzi. An implementation of basic argumentation components. In *Proceedings of the*

10th International Conference on Autonomous Agents and Multiagent Systems-Volume, pages 1307–1308, 2011.

[157] Henry Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.

[158] Henry Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.

[159] Henry Prakken. Models of persuasion dialogue. In I Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 281–300. 2009.

[160] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[161] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is ChatGPT a general-purpose natural language processing task solver? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1339—1384, 2023.

[162] Iyad Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[163] Ariel Rosenfeld and Sarit Kraus. Strategical argumentative agent for human persuasion. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, pages 320–328, 2016.

[164] Aurora Saibene, Michela Assale, and Marta Giltri. Expert systems: Definitions, advantages and issues in medical field applications. *Expert Systems with Applications*, 177:114900, 2021.

[165] Isabel Sassoon, Nadin Kökciyan, Elizabeth Sklar, and Simon Parsons. Explainable argumentation for wellness consultation. In *Proceedings of the International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 186–202, 2019.

[166] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing 36*, 2023.

[167] Marijn Schraagen and Floris Bex. Extraction of semantic relations in noisy user-generated law enforcement data. In *Proceedings of the 13th IEEE International Conference on Semantic Computing*, pages 79–86, 2019.

[168] Edward H Shortliffe. Mycin: A knowledge-based computer program applied to infectious diseases. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 66–69. American Medical Informatics Association, 1977.

[169] David Silver and Joel Veness. Monte-Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems 23*, 2010.

[170] E. I. Sklar, S. Parsons, and M. Davies. When is it okay to lie? A simple model of contradiction in agent-based dialogues. In *Proceedings of the First Workshop on Argumentation in Multiagent Systems*, 2004.

[171] Elizabeth I Sklar and Mohammad Q Azhar. Argumentation-based dialogue games for shared control in human-robot systems. *Journal of Human-Robot Interaction*, 4(3):120–148, 2015.

[172] Elizabeth I Sklar and Mohammad Q Azhar. Explanation through argumentation. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, pages 277–285, 2018.

[173] Elizabeth I Sklar, Simon Parsons, Zimi Li, Jordan Salvit, Senni Perumal, Holly Wall, and Jennifer Mangels. Evaluation of a trust-modulated argumentation-based interactive decision-making tool. *Journal of Autonomous Agents and Multi-Agent Systems*, 30(1):136–173, 2016.

[174] Mark Snaith, John Lawrence, Alison Pease, and Chris Reed. A modular platform for argument and dialogue. In *Proceedings of the Conference on Computational Models of Argument*, pages 473–474, 2020.

[175] Abonia Sojasingarayar. Seq2Seq AI chatbot with attention mechanism. Master's thesis, IA School Groupe GEMA, Boulogne-Billancourt, May 2020.

[176] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 244–257, 2009.

[177] Dan Sperber and Deirdre Wilson. *Relevance: Communication and cognition*. Harvard University Press, 1986.

[178] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. An overview of clinical decision support systems: benefits, risks, and strategies for success. *npj Digital Medicine*, 3(17), 2020.

[179] Yuqing Tang, Kai Cai, Peter McBurney, Elizabeth Sklar, and Simon Parsons. Using argumentation to reason about trust and belief. *Journal of Logic and Computation*, 22(5):979–1018, 2012.

[180] Yuqing Tang, Elizabeth Sklar, and Simon Parsons. An argumentation engine: Argtrust. In *Ninth International Workshop on Argumentation in Multiagent Systems*, 2012.

[181] Matthias Thimm. Tweety: A comprehensive collection of java libraries for logical aspects of artificial intelligence and knowledge representation. In *Fourteenth International Conference on the Principles of Knowledge*

Representation and Reasoning, pages 528–537, 2014.

[182] Matthias Thimm. Fargo-limited v1. 1.1. *ICCMA 2023*, page 21, 2023. https://helda.helsinki.fi/server/api/core/bitstreams/2e6d8485-fa34-48e5-9e24-eefe0cc1572c/content.

[183] Matthias Thimm. Harper++. 1.1. *ICCMA 2023*, page 27, 2023. https://helda.helsinki.fi/server/api/core/bitstreams/2e6d8485-fa34-48e5-9e24-eefe0cc1572c/content.

[184] Matthias Thimm, Federico Cerutti, and Mauro Vallati. Fudge v3. 2.8. *ICCMA 2023*, page 25, 2023. https://helda.helsinki.fi/server/api/core/bitstreams/2e6d8485-fa34-48e5-9e24-eefe0cc1572c/content.

[185] Matthias Thimm and Serena Villata. The first international competition on computational models of argumentation: Results and analysis. In *Artificial Intelligence*, volume 252, pages 267–294. Elsevier, 2017.

[186] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36, 2024.

[187] Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review*, 36, 2021.

[188] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[189] Gerard Vreeswijk. IACAS: An interactive argumentation system. User manual, version 1.0. Technical Report 94-03, University of Limburg, Department of Computer Science, 1994.

[190] Richard Wallace. *The elements of AIML style.* Alice AI Foundation, 2003.

[191] Richard S Wallace. The anatomy of A.L.I.C.E. In R. Epstein, G. Roberts, and G. Beber, editors, *Parsing the Turing Test*, pages 181–210. Springer, 2009.

[192] D. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* State University of New York Press, 1995.

[193] Douglas Walton. *Argumentation schemes for presumptive reasoning.* Routledge, 1996.

[194] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebas-

tian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[195] Joseph Weizenbaum. Eliza — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[196] Simon Wells and Chris A Reed. A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic*, pages 309–329, 2012.

[197] Andreas Xydis, Christopher Hampson, Sanjay Modgil, and Elizabeth Black. Enthymemes in dialogues. In *Proceedings of the Conference on Computational Models of Argument*, pages 395–402, 2020.

[198] Andreas Xydis, Christopher Hampson, Sanjay Modgil, and Elizabeth Black. Towards a sound and complete dialogue system for handling enthymemes. In *International Conference on Logic and Argumentation*, pages 437–456, 2021.

[199] Andreas Xydis, Christopher Hampson, Sanjay Modgil, and Elizabeth Black. A sound and complete dialogue system for handling misunderstandings. In *4th International Workshop on Systems and Algorithms for Formal Argumentation*, pages 19–32, 2022.

[200] Chunli Yan, Helena Lindgren, and Juan Carlos Nieves. A dialogue-based approach for dealing with uncertain and conflicting information in medical diagnosis. *Journal of Autonomous Agents and Multi-Agent Systems*, 32:861–885, 2018.

[201] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. https://arxiv.org/pdf/2303.18223, 2023.