Argunet – A software tool for collaborative argumentation analysis and research

David C. Schneider, Christian Voigt, Gregor Betz
Institute of Philosophy
Freie Universität Berlin
dc.schneid@googlemail.com, christian.voigt@denkartist.de, g.betz@philosophie.fu-berlin.de

Abstract

In this paper, we describe Argunet, a software tool for collaborative argumentation analysis. Its underlying theory, the theory of dialectical structures, defines argument relations in terms of locigo-sematical relations between propositions and thus gives rise to a new computational approach to the reconstruction of complex debates. We illustrate theoretical and practical advantages of the approach, describe Argunet's core algorithms and its support of a typical reconstruction workflow.

1 Introduction

In this paper, we describe Argunet[1], a software tool for collaborative argumentation analysis and research which is currently being developed by the authors, and its theoretical background. In section 2, we briefly introduce and motivate the theoretical framework. In section 3, a formal view on the framework is adopted in order to demonstrate its benefits for a computational approach to argumentation; core algorithms and elements of Argunet's data model are described. In section 4 we demonstrate how the program supports a typical workflow for argumentation reconstruction.

2 The theory of dialectical structures

According to our approach, single arguments are considered as premiss-conclusion structures.[2] Even though this structure is not necessarily explicit in everyday argumentation, every reason is clearly thought to back some claim and can thence be reconstructed as consisting of a conclusion backed by premisses.

Complex argumentations and debates are composed of many individual arguments between which different dialectical relations hold. In particular, arguments can support and attack each other. It is a core assumption of the theory of dialectical structures that these dialectical relations are determined by the logico-semantic relations between the arguments' premisses and conclusions. Correspondingly, an argument a supports (attacks) another argument b if and only if a's conclusion is equivalent (contrary) to one of b's premisses. The set of individual arguments plus the induced support and attack relations make up a debate's dialectical structure.

As figure 1 shows, argumentation analysis is a multi-level analysis, distinguishing the study of individual sentences, single arguments, and whole argumentations. Ontologically, argumentations are composed of single arguments which in turn consist of individual propositions, namely a conclusion plus premisses. The structures that are realized on the different levels are the dialectical structure on the top level, the argument- or inference-pattern on the single-argument level, and the logico-semantic structure of the sentences on the bottom level. As the dialectical structure depends on the structure of single arguments, the analysis of complex argumentations requires the analysis of single arguments. Reconstructing and investigating these presupposes, in turn, the analysis of individual sentences. Although the analysis of debates is therefore mainly carried out bottom-up, that is single arguments are reconstructed first, requiring logico-semantic analysis of their respective premisses and conclusions, before the overall dialectical structure is set up given the single arguments, this is not to say that insights into the dialectical structure cannot in turn feed back on the reconstruction of single arguments. Quite the contrary, it might be most helpful to consider the intended function of a single argument in its wider dialectical context when interpreting and reconstructing that very argument.

2.1 Advantages of the theory

This framework has different advantages that shall be sketched below.

First of all, the inclusion of both support and attack relations enables one to reconstruct and analyze controversial debates that include contradictory claims and reasonings.

Moreover, the general framework allows not only for the reconstruction of argumentation-trees, i.e. non-circular structures, but also circular argumentations of different sorts. This has, in particular, to be considered an advantage given that it is at least questionable whether every circular argumentation is fallacious; see e.g. [Walton, 2005; Betz, 2006].

Distinguishing between argument structure and argumentation, thirdly, allows one to process more complex argumentation without the structure of the debate becoming too complex, because, on the tertiary level of the dialectical structure,

[1] See www.argunet.org.

[2] For a more in-depth view on the theory of dialectical structures, see [Betz, 2005].

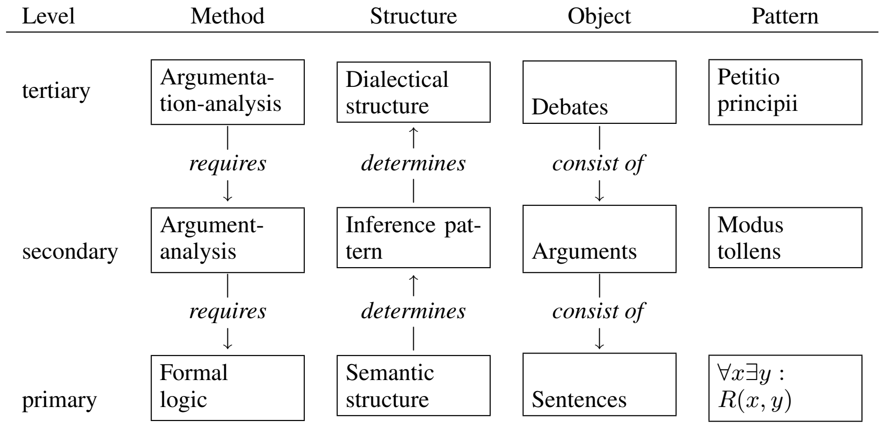

Figure 1: Multi-level analysis of debates.

the single-arguments are not represented in detail. That distinction allows, moreover, to apply already developed tools of argument analysis (e.g. [Tetens, 2004]) to single arguments, basing the theory of complex argumentation on well-established frameworks.

Moreover, deductivism, i.e. the reconstruction of arguments as deductively valid, can easily be incorporated into the theory of dialectical structures, yielding two further advantages.

Firstly, it provides argumentation analysts with inference-blueprints against which everyday arguments can be compared. Only against such a normative background can implicit premisses underlying informal reasonings be discovered and eventually made explicit, can discoveries during the reconstruction of debates. Here, the general idea behind deductivism is to use a minimal set of inference rules when reconstructing arguments. Thus understood, deductivism doesn't necessarily rely on classical logic, but on some set of inference rules. Inductive arguments do not really represent a problem for deductivism as further inference rules can always be made explicit as additional premisses.[3]

Secondly, the arguments' deductive structure plus the reduction of dialectical relations to logico-semantic relations between sentences facilitates the evaluation of dialectical structures, for they impose logical constraints on what subsets of arguments can be consistently asserted by a proponent in a debate at all. If, for example, argument a attacks argument b, a proponent cannot adhere to a in order to back a's conclusion while also accepting b's premisses without contradicting himself.

2.2 Research questions

The theory of dialectical structures provides opportunities to tackle old and new problems in argumentation theory, putting forward questions for further research. E.g.:

- What are necessary and sufficient conditions for the consistency of a proponent's stance in a complex debate?
- How can an important notions such as "burden of proof" or "petitio principii", which apparently lie beyond the scope of single argument analysis, be analyzed within a theory of dialectical structures?
- When can a thesis be said to be well justified given a certain dialectical structure?
- Does a debate's dialectical structure allow to infer rational strategies which proponents should pursue in order to reach their discursive aims?
- Are there characteristic macro-patterns whose realizations in debates reveal interesting features?

[3] See, e.g. [Groarke, 1999; Musgrave, 1999; Groarke, 2002]. For a critical view on deductivism, see [Govier, 1985; Johnson, 2000].

3 Algorithms and data model

In the following, a formal framework for the view on argumentation described above is sketched out. This allows us to adress computational issues as well as elements of Argunet's data model.

Let T be a set of arguments, $A \subseteq T \times T$ the attack relation and $U \subseteq T \times T$ the support relation. Then a dialectical structure is formally a triple $\tau = (T, A, U)$. Each dialectical structure has an isomorphic, directed graph $G_\tau = (V, E)$ with $V = T$ and a two-colored edge set $E = A \cup U$. Edges $(u, v) \in A$ are called "red edges" and represent attack relations; edges $(u, v) \in U$ are called "green edges" and represent support. The notion of the isomorphic graph G_τ is useful for formal and computational treatment of problems in argumentation theory; see, for example, [Betz, 2005]. Embeddings of G_τ in the plane serve as intuitively comprehensible visualizations of the underlying dialectical structure.

As introduced in section 2, one of the benefits of viewing arguments as premiss-conclusion-structures is that the relations between the arguments of a dialectical structure τ (and therewith the edges of G_τ) can be derived computationally from logico-semantic relations between propostions. In the following we describe how this derivation is performed in Argunet.

Let P be the set of propositions. Each $p \in P$ is a self-contained entity in the data model and stores the natural language content of the proposition as well as a unique numerical identifier. An argument is described as a sequence $\langle p_1, p_2, ..., p_n \rangle$ with $p_i \in P$. Within an argument, each proposition is associated with the role of premiss, assumption for the sake of the argument, preliminary conclusion or conclusion of the argument. Clearly, not all possible combinations of role associations are valid; an argument must have, for example, exactly one conclusion and at least one premiss. If each role is identified by a symbol, valid combinations of roles can be described by a context-free grammar and thus be checked by the computer.

This way of representation makes the sharing of propositions between arguments transparent to the computer and thus increases the formal expressiveness of the model. Additionally, the logico-semantic relations of equivalence and of contradiction between propositions are represented in the system as follows.

Trivially, equivalent propositions form equivalence classes. Thus a mapping $\epsilon : P \to \mathbb{N}$ of proposition identifiers to integers identifying equivalence classes is stored; a proposition p is equivalent to a proposition q iff $\epsilon(p) = \epsilon(q)$.

Contradictions are most efficiently described as holding between equivalence classes rather than individual propositions and are thus stored as a mapping $\gamma : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ of class identifiers to sets of identifiers of contrary classes.[4] γ is kept symmetric so that $q \in \gamma(p) \iff p \in \gamma(q)$ for arbitrary propositions p, q at all times. Clearly, a proposition p is contrary to a proposition q iff $\epsilon(p) \in \gamma(\epsilon(q))$ or, equivalently, $\epsilon(q) \in \gamma(\epsilon(p))$.

Given these representations of arguments and relations between propositions, the relations between arguments (and therewith the edges of the graph G_τ) can be computed efficiently:

Iterating over all arguments / vertices of G_τ, two mappings are construed: Firstly, $\chi : P \to \mathcal{P}(V_G)$ mapping propositions p to sets of vertices / arguments that contain p in the role of a conclusion, i.e. $p \in \chi(v)$ iff v's argument concludes to p. Secondly, $\psi : P \to \mathcal{P}(V_G)$ mapping propositions p to sets of vertices / arguments that contain p in the role of a premiss.

For the construction of the graph edges E, denote by χ_K the key set[5] of χ and, similarly, by ψ_K the key set of ψ. Let $(p, q) \in \chi_K \times \psi_K$ be an element the cross product of the two key sets. A red edge (u, v) is added to E iff

$$u \in \chi(p) \ \wedge \ v \in \psi(q) \ \wedge \ \epsilon(p) \in \gamma(\epsilon(q))$$

and, respectively, a green edge is added iff

$$u \in \chi(p) \ \wedge \ v \in \psi(q) \ \wedge \ \epsilon(p) = \epsilon(q).$$

4 Reconstructing with Argunet—Naturally and formally

In this section we give an overview of the use of Argunet in practice. From the preceding formalism and the treatment of deductivism in section 2, the impression may have arisen that the program is primarily aimed at formal logicians. Thus we first want to clarify which kinds of formalism Argunet supports and which kinds it requires the user to adopt.

Firstly, it should be pointed out that in Argunet propositions are formulated in natural language. Secondly, while Argunet does impose a premiss-conclusion-structure on arguments, there are no restrictions on the inference patterns leading from the premisses to conclusions. Thus the program is not based on classical formal logic, nor is it restricted to deductive reconstruction. Our approach recognizes the importance of natural language reconstruction and topic-specific inference rules; both are fully supported by Argunet.

At the same time, however, we think, that for a reconstruction to be a disputable analysis at all, there needs to be some kind of formalization. Additionally, *computational* argumentation analysis should benefit from the computer's ability to *compute* rather than just use it as a drawing tool for fancy presentation of debates. So the question is not, whether formalism makes sense in argumentation analysis, but rather which kind of formalism allows us to get a theoretical grip on arguments and their dialectical roles in debates without getting in the way of the naturalness of reconstruction.

One rigorous way of formalization supported, but not presupposed by the program, is deductive reconstruction of arguments. It is supplementary to but not required by the theory of dialectical structures which is the formalism at the heart of Argunet. This approach does not only give rise to new theoretical and computational possibilities, as pointed out in sections 2 and 3. It has practical advantages for the reconstruction of debates as well.

4.1 User interface and workflow

From a user's point of view, two characteristics set our approach apart from other argument visualization software (see, e.g. [van den Braak *et al.*, 2006]): Firstly, while most tools use a sentence-based reconstruction of dialectical structures, Argunet supports a multi-level, argument-based reconstruction. Secondly, the relations between arguments are derived from the logico-semantic relations between propositions. In the following we focus on the practical rather than the theoretical advantages of those features and give an overview of Argunet's interface.

In Argunet, the user primarily works in two editors, the *graph editor* and the *argument editor* (see figure 1). They support a common four step workflow of debate reconstruction:

(1) Sketching in the graph editor Typically, reconstruction begins by sketching out the main ideas of the arguments and their crucial relations. This first sketch may still be imprecise or flawed, but it helps to save time and keep track of the complete picture.

Argunet has a sketch mode allowing the user to place arguments in the graph and enter short descriptions that are displayed in the graph as well. Moreover, preliminary argument relations can be illustrated by dashed "sketched edges" without already defining semantic relations between individual propositions. Sketch mode is also useful for presentation as it allows to create very concise versions of debate graphs.

[4] $\mathcal{P}(X)$ denotes the power set of a set X.

[5] i.e. the subset of the domain that is actually mapped to the image

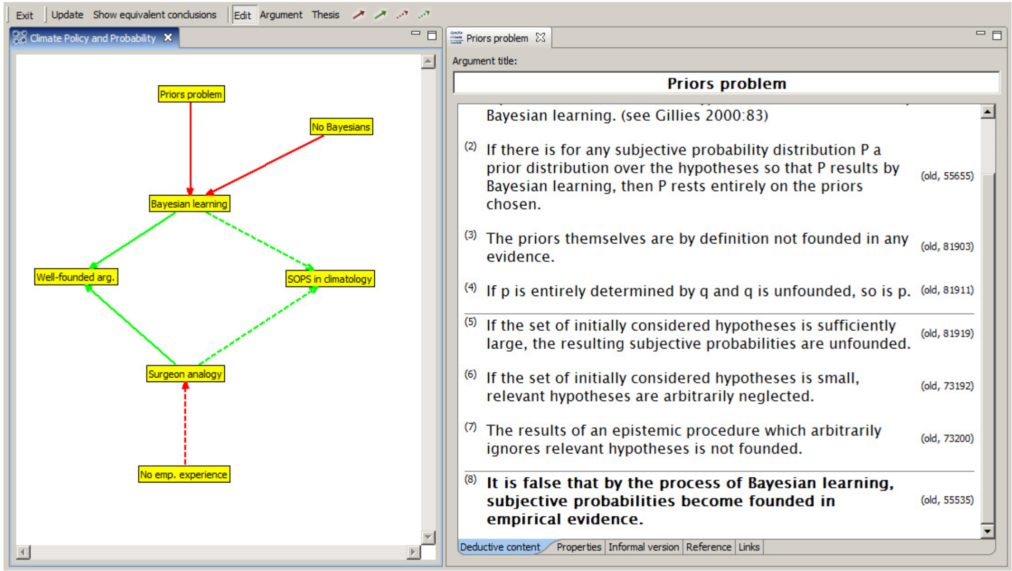

Figure 2: Screenshot of the current development version. Graph editor on the left, argument editor on the right.

Note that this is only possible in an approach which treats arguments as entities of their own.

(2) Reconstructing arguments In Argunet an argument is technically a list of propositions. Hence the argument editor is basically operated like the list editors in widespread text processors. By clicking on a proposition's numeral, the user can change its role in the argument (e.g. premise, conclusion, etc.). Since each proposition is an entitiy of its own in the data model, propositions can also be imported from the pool. Several types of additional information such as references or hyperlinks to files and online resources can be stored with each argument.

(3) Defining semantic relations There are several ways of defining semantic relations between propositions. One is by "drawing" a red or green edge in the graph editor from one argument vertex u to another vertex v. Argunet then considers the conclusion of u to be the first relatum. A dialog-box displaying argument v is presented from whose premisses the second relatum can be selected. Since a relation between a conclusion and a premiss has been established, at least one edge between u and v appears in the graph. In contrast to the sketched edges, these "semantic edges" are drawn as a solid line. If the relata are members of non-trivial equivalence classes, the new relation may induce other edges; they are computed by the program and appear in the graph view automatically.

(4) Organizing complex debates When debates are complex it is crucial that the argumentation is visualized in a structured and informative manner. In the following we give three examples of the instruments Argunet offers for handling complexity.

Firstly, if a graph becomes too large it may be appropriate to split up its visualization into different parts. An Argunet project can contain an infinite number of views[6] on a debate graph or any subgraph of it. Many things can be expressed by using multiple graph views: For example, if the user wants to demonstrate how a debate has changed over time she can use them to visualize different states. Since each graph view is based on the same pool of propositions, semantic relations and arguments, a complex debate with many views is automatically kept consistent and remains easy to maintain. This is a major practical advantage of the theory of dialectical structures.

Secondly, it may be useful to visualize how a debate is composed of many relatively independent subdebates. This is supported by visual node-grouping tools in the graph editor.

Thirdly, in complex debates it is often important to know the proponents of the arguments. In Argunet they can be visualized by user defined color codes for argument vertices.

4.2 Architecture and technology

Argunet can be used locally on a single computer as well as online in collaborative mode, allowing multiple clients to work together on the same debate in real time over the internet. Content can easily be copied back and forth between local and online debates. A sophisticated user management allows clients to invite others into their projects as editors or viewers. Thus Argunet can be used as a tool for online debating, for collaborative research and reconstruction projects or as an e-learning application. A prototype of the program has been successfully employed in philosophy classes for over a year. Arguments reconstructed online by students from home were discussed and semantically connected in class.

[6] i.e. embeddings in the plane, in the language of graph theory

Argunet is an open-source software written in the Java language (J2SE5) to ensure platform independence. The client program is based on the Eclipse Rich Client Platform ([des Rivieres and Beaton, 2006]) and thus has a flexible and modern user interface with a native platform look-and-feel. Persistence is implemented on top of the Db4O object oriented database ([Grehan, 2006]). A plugin mechanim ensures Argunet's extensibility.

5 Outlook

In our paper we described the underlying theory as well as the core features of our argumentation tool Argunet. As research questions in the theory of dialectical structures have already been posed in section 2.2, this brief outlook focuses on our future plans for the software.

At the time of writing Argunet is under development. A public beta testing phase is scheduled to begin in early 2007. Argunet users will be encouraged to use the server at the Freie Universität Berlin in order build up a searchable community database of arguments and debates. A considerable stock of debates that has already been reconstructed with the software prototype will be the foundation of the database. Backed by a large data set, Argunet will be the ideal testbed for argumentation theoretic algorithms. To ensure compatibility with other programs we plan to engage in the development of XML formats. Browser based presentation features that make debates reconstructed with Argunet accessible to a wider public are under development.

References

[Betz, 2005] Gregor Betz. The vicious circle theorem – a graph-theoretical analysis of dialectical structures. *Argumentation*, 19(1):53–64, March 2005.

[Betz, 2006] Gregor Betz. Petitio principii and circular argumentation as seen from a theory of dialectical structures. in preparation, 2006.

[des Rivieres and Beaton, 2006] Jim des Rivieres and Wayne Beaton. Eclipse platform technical overview. whitepaper, International Business Machines Corp., 2006.

[Govier, 1985] Trudy Govier. *A Practical Study of Argument*. Wadsworth Publishing Company, 1985.

[Grehan, 2006] Rick Grehan. Complex object structures, persistence, and db4o. whitepaper, db4objects Inc, 2006.

[Groarke, 1999] Leo Groarke. Deductivism within pragma-dialectics. *Argumentation*, 13:1–16, 1999.

[Groarke, 2002] Leo Groarke. Johnson on the metaphysics of argument. *Argumentation*, 16:277–286, 2002.

[Johnson, 2000] Ralph Johnson. *Manifest Rationality. A Pragmatic Theory of Argument.* LEA, 2000.

[Musgrave, 1999] Alan Musgrave. How to do without inductive logic? *Science and Education*, 8, 1999.

[Tetens, 2004] Holm Tetens. *Philosophisches Argumentieren*. Beck, 2004.

[van den Braak *et al.*, 2006] Susan W. van den Braak, Herre van Oostendorp, Henry Prakken, and Gerard A.W. Vreeswijk. A critical review of argument visualization tools: do users become better reasoners? In *6th Workshop on Computational Models of Natural Argument (CMNA VI)*, 2006.

[Walton, 2005] Douglas Walton. Begging the question in arguments based on testimony. *Argumentation*, 19:85–113, 2005.