Complex arguments in Adpositional Argumentation (AdArg)

Gobbo, F.; Benini, M.; Wagemans, J.H.M.

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 5th Workshop on Advances in Argumentation in Artificial Intelligence 2021

License
CC BY

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Complex Arguments in Adpositional Argumentation

Federico Gobbo¹, Marco Benini² and Jean H.M. Wagemans³

¹University of Amsterdam, Spuistraat 134, NL-1012 VB Amsterdam, The Netherlands
²University of Insubria, via Valleggio 11, IT-22100 Como, Italy
³University of Amsterdam, Spuistraat 134, NL-1012 VB Amsterdam, The Netherlands

Abstract
Adpositional Argumentation (AdArg) is a new method for annotating argumentative discourse that represents linguistic and pragmatic information in argumentative adpositional trees. In this paper, we explain how the representation of claims and individual arguments provide the building blocks for more complex argumentation structures. We illustrate the abstract trees representing the systematic possibilities of a claim (one statement), minimal argument (one conclusion, one premise), convergent argumentation (one conclusion, multiple premises), as well as serial argumentation, when the same linguistic material plays the double role of the premise of a given argument and the conclusion of a subargument.

Keywords
computational argumentation, argumentation theory, argumentation structure, complex argumentation

1. Introduction

Developing tools and models for annotating natural argumentative discourse requires a formalization of linguistic material. A major challenge in this endeavor is to find the right balance between, on the one hand, the level of linguistic detail to be incorporated in the tool or model and, on the other hand, its robustness from a formal point of view. In many approaches, the concessions are made on the linguistic side. The tools based on Walton’s argumentation schemes [1] and the Toulmin model [2], for instance, produce selective representations of premises and conclusions but they do not consider a great many linguistic items relevant for identifying the fabric of the argumentation. Computational models suitable for argument mining, for instance those based on Dung’s seminal paper on abstract argumentation [3], usually abstract away from linguistic details in representation the argumentation [4, 5].

Adpositional Argumentation (AdArg) is a new method for annotating argumentative discourse that aims to integrate formal robustness and linguistic accuracy, without imposing predefined structures onto the original discourse [6, 7]. Different from existing tools and models, its procedure for reconstructing the fabric of a given argumentative text leaves the linguistic
material completely intact, enabling the analyst to hide or highlight aspects relevant for their purposes. In earlier work on AdArg, it is explained how the linguistic representation framework of Constructive Adpositional Grammars (CxAdGrams) [8] can be combined with the argument categorisation framework of the Periodic Table of Arguments (PTA) [9, 10].

The general goal of AdArg is to explicate the process of annotating natural argumentative discourse through a detailed representation of linguistic as well as pragmatic information. We argue that the formal representation of the dynamics and the results of such a process eventually facilitates textual analysis for humans as well as machines. So far, we have focused on describing how our method can represent individual arguments of various types, while acknowledging that actual argumentative discourse usually consists of a more complex whole of conclusions and premises.

In this paper, we stay at the highest level of abstraction possible, hiding the linguistic details in the small triangles (△; see all Figures below). In particular, we first explain how the representations of individual arguments provide the building blocks for complex argumentation structures. Then, we show how our method deals with complex argumentation starting from the notions of ‘minimal argument’ and ‘argument form’ described in the PTA, already presented in previous publications [6, 7]. Finally, we illustrate the abstract trees representing the systematic possibilities of a claim (one statement), a minimal argument (one premise, one conclusion), and convergent argumentation (multiple premises, one conclusion). Moreover, we show how to conceive the abstract tree of serial argumentation, i.e., when the same linguistic material plays the double role of the premise of a given argument and the conclusion of a subargument.

2. Representing claims and minimal argument forms

In the theoretical framework of the PTA, a ‘minimal argument’ is defined as a combination of two statements, one of which functions as the conclusion (ṣ) and the other as the premise (π). The ‘argument form’ of a minimal argument is the specific configuration of the subjects and the predicates of these two statements, after the analyst has reconstructed their original linguistic expression into the retrogressive normal form (conclusion, because premise) [11]. Table 1, adapted from [6], offers a synthetic overview of the four normal forms of minimal arguments, one for each Quadrant (Q) [9].

<table>
<thead>
<tr>
<th>Quadrant (Q)</th>
<th>conclusion (ṣ)</th>
<th>premise (π)</th>
<th>retrogressive normal form</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>a is X</td>
<td>a is Y</td>
<td>a is X, because a is Y</td>
</tr>
<tr>
<td>β</td>
<td>a is X</td>
<td>b is X</td>
<td>a is X, because b is X</td>
</tr>
<tr>
<td>γ</td>
<td>q is T</td>
<td>r is T</td>
<td>q is T, because r is T</td>
</tr>
<tr>
<td>δ</td>
<td>q is T</td>
<td>q is Z</td>
<td>q is T, because q is Z</td>
</tr>
</tbody>
</table>

In the actual discourse, a minimal argument can also be expressed in a progressive form (premise, therefore conclusion) [12, p. 33]. Here we stick to the retrogressive form (conclusion, because premise), indicated by: →. Formally, they are equivalent.
When annotating argumentative discourse, the analyst may find various constellations of statements relevant to include in the representation. The first and simplest possibility is the presence in the discourse of a ‘claim’, i.e., a single statement not supporting another one. By default, such a statement is interpreted as a conclusion ($\sigma$) – Figure 1, adapted from [7], left. Regarding minimal arguments, which consist of two statements, Figure 1 (center, right) shows that the form of the arg-adtrees depends on the Quadrants where they actually belong; each of them corresponds to the retrogressive normal forms already seen in Table 1.

The level of natural language data is always at the bottom layer of the leaves in the arg-adtree, conventionally indicated by ‘txt’, which is substantiated in Figure 1 by the subjects $(a, b)$ and predicates $(X, Y)$. The small triangles ($\triangle$) indicate that it is possible to unfold the linguistic structure of that data using the theoretical framework of Constructive Adpositional Grammar [8, 6].

The unfolding of the minimal argument forms shows the fundamental structures of abstract arg-adtrees. Figure 2 illustrates that the structure of Alpha, Beta, and Gamma arguments consists of different configurations of subjects $(a, b)$ and predicates $(X, Y)$ of the propositions $(q, r)$ involved, while the premise of Delta arguments has a predicate $(Z)$ attributed to the conclusion, which appears in the arg-adtree as quoted $(q)$ [9].

---

2The “is true” (is $\top$) that is added to the second-order arguments of the Gamma and Delta quadrants is not a logical value but expresses in natural language the standardized doxastic commitment of the arguer to the acceptability of the proposition.
3. Representing complex arguments

Unlike minimal arguments, complex arguments have either multiple premises or use the premise (\(\pi\)) as the conclusion (\(\sigma\)) for a subargument, i.e., an argument depending on the main one: the first ones are right-growing, called convergent, while the second ones are called serial. Serial arguments are so called because they provide a complex argument which is, literally, a chain of minimal arguments. Convergent arguments consist of one conclusion (\(\sigma\)) supported by the finite multiple premises (\(\pi_1, \pi_2, ..., \pi_n\)). The conclusion may be supported by different types of arguments, depending on the specific minimal argument form identified when analyzed in isolation.

In serial arguments, the same linguistic element (txt) plays the role both of the premise (\(\pi_1\)) of the hierarchically highest argument and of the conclusion (\(\sigma_2\)) of the subargument. To represent this double function of txt, the notation \(\omega(\pi_1, \sigma_2)\) is used, in which the \(\omega\) graphically represents the two halves of a chain ring. Specifically, \(\omega(\pi_1, \sigma_2)\) represents the reason why the two arguments can be joined. The intuition is that \(\pi_2 \implies \sigma_2 = \text{txt} = \pi_1\), which in turn implies \(\sigma_1\). The verb implies clearly shows that the logic of the arguer comes in here: whatever notion of implication she uses, it has to interpolate \(\pi_2\) and \(\sigma_1\) through txt. For example, if the arguer uses classical logic, implies would be the Aristotle’s implication \(\supset\), thus \(\omega(\pi_1, \sigma_2)\) stands for the logical tautology \((\pi_2 \supset X) \land (X \supset \sigma_1)\), explaining how the arguments meet on \(X\), which gets instantiated to txt. However, the arg-tree representation is pre-logical, and it does not force any specific way to interpret \(\omega\). In fact, whatever notion of implication \(\supset\) and meet\(^3\) \(\land\) the arguer uses, it has to support the fact that \((\pi_2 \supset \text{txt}) \land (\text{txt} \supset \sigma_1)\). Hence, the argumentation conveys also a snapshot of the logic of the arguer, and it is contained in the representation tree. This marks a huge difference with most other approaches, in which the logic is presumed.

4. Discussion and further work

In this paper we have described how AdArg represents various constellations of argumentative statements expressed in natural language. The representation results in an annotation of the discourse that reveals argumentation structures and patterns. This annotation is pre-logical: it shows exactly where the logic comes in, and thus, in the eyes of the argumentation analyst, it

\(^3\)The name is not by chance, reminding to the interpretation of conjunction in algebraic lattices.
makes evident the point of attack of the way of reasoning, i.e., the logic of the arguer. To explain this aspect in full, an empirical analysis that comprises linguistic and pragmatic elements is needed; this is left as a further work. Moreover, the proposed representation shows how the logic of the arguer can be partially inferred from the used arguments, specifically by showing how she believes arguments can be validly connected in sequences.

We claim that a rigorous and fine-grained annotation of arguments expressed in natural language can lead to the building of corpora suitable for processing in terms of argument mining and eventually Explainable AI.

References