Complex arguments in Adpositional Argumentation (AdArg)

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

Publication date
2021

Document Version
Final published version

Published in
AI^3 2021 : Advances in Argumentation in Artificial Intelligence

Complex Arguments in Adpositional Argumentation

Federico Gobbo[1], Marco Benini[2] and Jean H.M. Wagemans[3]

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

Marcello D'Agostino, Fabio Aurelio D'Asaro and Costanza Larese, eds, 5th Workshop on Advances In Argumentation In Artificial Intelligence (AI^{3}2021), *December* 01 − 03, 2021, *Milan, IT*
f.gobbo@uva.nl (F. Gobbo); marco.benini@uninsubria.it (M. Benini); j.h.m.wagemans@uva.nl (J. H.M. Wagemans)
https://federicogobbo.name/ (F. Gobbo); https://marcobenini.me/ (M. Benini); https://uva.academia.edu/JeanHMWagemans (J. H.M. Wagemans)
0000-0003-1748-4921 (F. Gobbo); 0000-0003-4680-5982 (M. Benini); 0000-0001-9304-5766 (J. H.M. Wagemans)

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 (σ) 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].[1] Table 1, adapted from [6], offers a synthetic overview of the four normal forms of minimal arguments, one for each Quadrant (Q) [9].

Table 1
Overview of abstract minimal arguments in the PTA

Quadrant (Q)	conclusion (σ)	premise (π)	retrogressive normal form
α	a is X	a is Y	a is X, because a is Y
β	a is X	b is X	a is X, because b is X
γ	q is $\top$	r is $\top$	q is $\top$, because r is $\top$
δ	q is $\top$	q is Z	q is $\top$, because q is Z

[1] 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: $\longrightarrow$. 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 (σ) – 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.

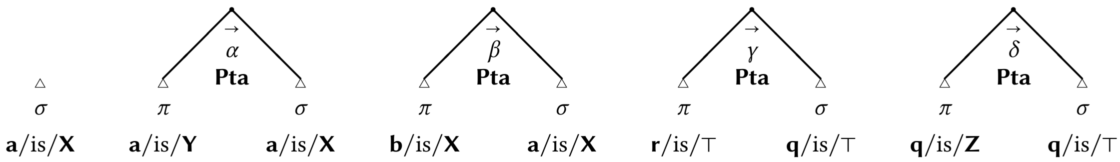

Figure 1: Claims (left) and abstract trees of minimal argument forms (center, right)

The level of natural language data is always at the bottom layer of the leaves in the arg-adtree, conventially indicated by ‘txt’, which is substantiated in Figure 1 by the subjects (a, b) and predicates (X, Y).[2] The small triangles (△) 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].

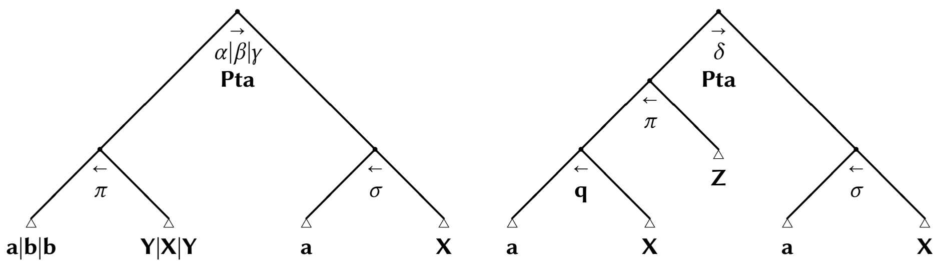

Figure 2: Basic abstract trees of the Alpha, Beta and Gamma (left), and Delta (right) arguments

[2]The “is true” (is ⊤) 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 (π) as the conclusion (σ) 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 (σ) supported by the finite multiple premises (π_1,π_2,...,π_n). The conclusion may be supported by different types of arguments, depending on the specific minimal argument form identified when analyzed in isolation.

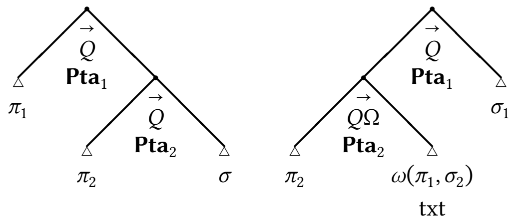

Figure 3: The abstract arg-adtrees of complex arguments: convergent (left) and serial (right)

In serial arguments, the same linguistic element (txt) plays the role both of the premise (π_1) of the hierarchically highest argument and of the conclusion (σ_2) of the subargument. To represent this double function of txt, the notation $\omega(\pi_1, \sigma_2)$ is used, in which the ω 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 π_2 *implies* $\sigma_2 = \text{txt} = \pi_1$, which in turn *implies* σ_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 π_2 and σ_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 $(\pi_2 \supset X) \wedge (X \supset \sigma_1)$, 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 ω. In fact, whatever notion of *implication* $\supset$ and *meet*[3] $\wedge$ the arguer uses, it has to support the fact that $(\pi_2 \supset \text{txt}) \wedge (\text{txt} \supset \sigma_1)$. 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

[1] D. N. Walton, C. Reed, F. Macagno, Argumentation Schemes, Cambridge University Press, Cambridge, 2008.

[2] S. Toulmin, The uses of argument, Cambridge University Press, Cambridge, 1958.

[3] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–357. doi:10.1016/0004-3702(94)00041-X.

[4] S. Modgil, K. Budzynska, J. Lawrence (Eds.), Computational Models of Argument, 2018.

[5] M. Lippi, P. Torroni, Argumentation mining: State of the art and emerging trends 16 (2016). URL: https://doi.org/10.1145/2850417. doi:10.1145/2850417.

[6] F. Gobbo, J. H. M. Wagemans, A method for reconstructing first-order arguments in natural language, in: P. Dondio, L. Longo (Eds.), Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence (AI*IA 2018), volume 2296, CEUR-WS.org, University of Trento Italy, 2019, pp. 27–41. URL: http://ceur-ws.org/Vol-2296/AI3-2018_paper_4.pdf.

[7] F. Gobbo, M. Benini, J. H. M. Wagemans, Annotation with adpositional argumentation, Intelligenza Artificiale 13 (2019) 155–172. doi:10.3233/IA-190028.

[8] F. Gobbo, M. Benini, Constructive Adpositional Grammars, Cambridge Scholars Publishing, Newcastle upon Tyne, 2011.

[9] J. H. M. Wagemans, Four basic argument forms, Research in Language 17 (2019) 57–69. doi:10.2478/rela-2019-0005.

[10] J. H. M. Wagemans, Constructing a Periodic Table of Arguments, in: P. Bondy, L. Benacquista (Eds.), Argumentation, Objectivity, and Bias: Proceedings of the 11th International Conference of the Ontario Society for the Study of Argumentation, OSSA, Windsor, ON, 2016, pp. 1–12.

[11] J. H. M. Wagemans, Argument Type Identification Procedure (ATIP), 2020. URL: www.periodic-table-of-arguments.org/argument-type-identification-procedure.

[12] F. H. van Eemeren, A. F. Snoeck Henkemans, Argumentation. Analysis and Evaluation, Routledge, London, 2016.