Adpositional Argumentation (AdArg)

A new method for representing linguistic and pragmatic information about argumentative discourse

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

Published in:
Actes 13èmes Journées d'Intelligence Artificielle Fondamentale (JIAF 2019)

Adpositional Argumentation (AdArg): A new method for representing linguistic and pragmatic information about argumentative discourse

Federico Gobbo[1] **Jean H.M. Wagemans**[2]

[1] ACLC, University of Amsterdam, The Netherlands / StudiUm, University of Turin, Italy
[2] ACLC, University of Amsterdam, The Netherlands
F.Gobbo@uva.nl J.H.M.Wagemans@uva.nl

Résumé

Cet article décrit et illustre l'utilisation de *Adpositional Argumentation* (AdArg), une méthode formelle nouvelle qui permet à l'analyste du discours argumentatif de représenter des informations linguistiques et pragmatiques de manière très détaillée et quand même flexible. L'article explique d'abord les points de départ théorétiques de la méthode, qui associe le cadre de représentation linguistique *Constructive Adpositional Grammars* (CxAdGrams) et le cadre de classification des arguments *Periodic Table of Arguments* (PTA). Il détaille ensuite les étapes de base de la méthode et les illustre en expliquant comment construire un 'argumentative adpositional tree' (ou 'arg-adtree') d'un exemple concret d'argument.

Abstract

This paper describes and illustrates the use of *Adpositional Argumentation* (AdArg), a new formal method that enables the analyst of argumentative discourse to represent linguistic and pragmatic information in a highly detailed and yet flexible way. It first explains the theoretical starting points of the method, which is a combination of the linguistic representation framework of *Constructive Adpositional Grammars* (CxAdGrams) and the argument classification framework of the *Periodic Table of Arguments* (PTA). It then lays out the basic steps of the method and illustrates them by explaining how to build a so-called 'argumentative adpositional tree' (arg-adtree) of a concrete example of an argument.

1 Introduction

Scholars in the fields of Argumentation Theory and Rhetoric have developed a great many insights concerning the way in which people support their points of view with arguments. These linguistic and pragmatic insights pertain to the nature and constituents of various types of arguments, the structure of different genres of argumentative discourse, as well as the stylistic features of such discourse. In combination with normative standards regarding the validity, reasonableness, and effectiveness of argumentation, this knowledge is used for providing theoretically informed analyses and evaluations of argumentative texts and discussions – see, e.g., [4]

So far, researchers in the fields of Artificial Intelligence and Computational Argumentation have used only a small part of this *cornu copiae* of knowledge. Their computational models of argument and tools for argument mapping, argument mining and computer-aided decision-making usually operate on the abstract level of complete propositions and the interactions between them. This goes, for example, for approaches inspired on Dung's abstract argumentation frameworks, which study sets of atomic arguments and their interrelations. But it also applies to approaches that take Walton's argument schemes as a point of departure, in which an argument scheme is taken to consist of a conclusion and a set of premises – see, e.g., [10].

One of the factors that explains the lack of interaction between the two research areas is that the insights about argumentation and rhetoric developed by scholars from the humanities, although profound and detailed, are mainly informal in nature. Therefore, in order to be used by more formally oriented researchers, these insights need to be translated in such a way that formalizing them does not result in a decrease of the richness of the information that can be represented or in a loss of relevant details.

Adpositional Argumentation (AdArg) is a constructive linguistic approach to argumentation that is aimed at formalizing the insights developed within Argumentation Theory and Rhetoric so as to enable their integration into the models and frameworks developed within Artificial Intelligence and Computational Argumentation. One of the first achievements of this approach is the development of a method for the formal representation of linguistic and pragmatic information concerning arguments expressed in natural language. The method produces so-called 'argumentative adpositional trees' (or 'arg-adtrees'), which enable the analyst not only to represent sentences on the morphosyntactic level, but also to include information regarding the argumentative function of their constituents. The parsing largely depends on the natural language in use, while the pragmatic information is language-independent to a large extent.

In this paper, we explain the basic characteristics of this representation method and illustrate its use by constructing an arg-adtree of an argument expressed in natural language. For the sake of brevity and clarity, in our example we focus on representing the conclusion and the premise of a single argument on the level of the individual words. As we intend to demonstrate, the method enables the analyst to identify in a very detailed way those linguistic elements that have a pragmatic function in the argumentation. At the same time, it is highly flexible in that the analyst is free to choose which details to show and hide, according to her needs. In particular, the representation of the argument in an arg-adtree may prepare the ground for an assessment of its quality, helping the analyst to identify in a very precise way the 'points of attack' of the argument under scrutiny.

Our method for representing arguments in adtrees results from a combination of two theoretical frameworks. Its basic characteristics are derived from *Constructive Adpositional Grammars* (CxAdGrams), a formal linguistic framework developed by Gobbo and Benini [6] that employs adpositional trees for the purpose of representing natural language. The addition of a layer of pragmatic information to these adtrees is achieved by using the *Periodic Table of Arguments* (PTA), an argument classification framework developed by Wagemans [13, 17] that is especially suitable for formal linguistic and computational approaches to argument. In the following, we explain the theoretical starting points of CxAdGrams (Section 2) and the PTA (Section 3). We then lay out the basic steps of the method and illustrate them by explaining how to construct an arg-adtree of a concrete example of an argument expressed in natural language (Section 4). We conclude the paper with a short reflection on further research and applications (Section 5).

2 The linguistic representation framework

The theoretical framework of *Constructive Adpositional Grammars* (CxAdGrams) is the result of the application of constructive mathematics to the adpositional paradigm in linguistics. We first elucidate this framework by explaining the meaning of the key terms 'constructive' and 'adpositional' in this particular context. Then, we explain the central notion of 'adpositional tree' and illustrate its characteristics by means of an example.

Constructive mathematics is an approach to mathematics that is premised on the idea that regarding the formulas of a theorem, the information content of any statement should be strictly preserved – see Bridges and Richman [2]. There is a tradition of using constructive mathematics to formally represent natural languages, starting from the work of Adjukiewicz [1] and Church [3]. Of the various constructive models in mathematical and computational linguistics developed so far, CxAdGrams specifically are based on topos theory. It thus permits to use Grothendieck's topoi as the mathematical instrument to formalize natural languages and their regularities, both intra a single language and between two or more natural languages in comparison.

The adpositional paradigm in linguistics follows the idea that each pair of linguistic elements can be conveniently described in terms of asymmetrical relations, that is, in such a way that their arrangement cannot be reversed. Thus, given a pair of morphemes, words or expressions, one element 'governs' the other, and consequently, the latter element 'depends' on the former. A very basic example is the phrase *children play*, which has the verb *play* as the governing element (*gov*) and the noun *children* as the dependent element (*dep*). The hierarchical relation between a pair of linguistic terms is conventionally called an 'adposition' and can be pictured in a so-called 'adpositional tree' (or 'adtree').

CxAdGrams, then, is the result of taking a constructive mathematical approach to representing natural language in adpositional trees. More specifically, the formation of an 'adpositional grammar' (or 'adgram'), a set of rules for building adtrees that is admissible within a given natural language, follows certain meta-rules that are described in terms of Grothendieck topoi. As a result, the adtrees produced within CxAdGrams do not only represent natural language expressions in the form of recursive trees but can also be interpreted as formulas – which means that they are suitable for the purpose of natural language processing (for an example, see Figure 4 below). [1]

We now turn to explaining the notion of 'adpositional tree' in more detail. A minimal adtree consists of a pair of linguistic elements and their relation, expressed in terms of their adposition. The governing element (*gov*) is conventionally put on the right leaf at the bottom of the rightmost branch, while the dependent element (*dep*) is put on the left leaf at the bottom of the leftmost branch. The adposition (*adp*), which represents the relation between the governor and the dependent, is depicted as a hook under the bifurcation of the two branches. Under both elements and their adposition, a grammar character (*gc*) is placed. In order to

1. The linguistic and formal rules of CxAdGrams are not discussed here for reasons of conciseness. For a comprehensive presentation of this approach to linguistic analysis, see [6]. The formal model is presented in Appendix B of that work.

illustrate these concepts, we picture in Figure 1 the adtree of the phrase *children play* that was mentioned above.

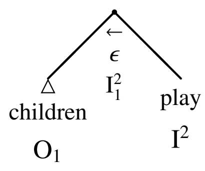

FIGURE 1 – The adtree of the phrase *children play*

In this example, the governor is substantiated by *play*, the dependent by *children*, and the adposition by an epsilon (ϵ), indicating that there is a syntactic relation between the two words. In general, a triangle (△) indicates the possibility of recursion, i.e., the fact that another adtree can be appended to each leaf recursively. In this particular case, it indicates the possibility of representing morphological information regarding the word *children*, which is irrelevant for the present purposes but illustrates the fact that the analyst can hide or show details according to her needs.

The theoretical framework of CxAdGrams uses five different grammar characters, which are represented by five vowels (A, E, I, O, U). Table 1 explains their meaning – adapted from [6, 41].

gc	*name*	*function*	*examples*
A	adjunctive	modifier of *O*	adjectives, articles, determiners
E	circumstantial	modifier of *I*	adverbs, adverbial expressions
I	verbant	valency ruler	verbs, interjections
O	stative	actants	nouns, pronouns, name-entities
U	underspecified	transferer	prepositions, derivational morphemes

TABLE 1 – The meaning of grammar characters in adtrees

The overall shape of an adtree is mainly defined by verbants – typically, verbs (see Table 1). Their grammar characters (I) and those of the correlated nominal expressions (O) may show additional parameters. In the case of verbants, an apex indicates the verbal valency (*val*), i.e., the number of actants that are potentially involved in the activity described by the verb. [2] The I^2 in our example indicates that the verb *play* is bivalent (*val* = 2), because semantically it implies a player (the first actant) as well as a game or a musical instrument (the second actant). [3]

In the case of the nominal expressions correlated with the verbant, a pedix indicates the number by which they are identified. In our example, *children* acts as the first actant (O_1), while the second actant has remained implicit.

Finally, the information expressed in the complete adtree is summarized by the grammar character I^2_1 under the hook. Here, again, the apex indicates the valency value (*val*) of the verb, while the pedix indicates the number of actants (*act*) present in the sentence. The former is bigger than the latter (*val* = 2 and *act* = 1), which means that the verb is only partially saturated.

We now discussed most of the basic aspects of linguistic adtrees. The meaning of arrows, such as the left arrow (←) above the epsilon (ϵ) in Figure 1, will be explained in Section 4. Given our current purpose of showing how CxAdGrams can be applied to the pragmatic relation between premise and conclusion in an argument, we first turn to expounding the theoretical framework of the PTA, a formal linguistic approach to argument classification.

3 The argument classification framework

The *Periodic Table of Arguments* (PTA) is a classification of argument that integrates the traditional dialectical accounts of argument schemes and fallacies as well as the rhetorical accounts of logical, ethotic, and pathetic means of persuasion into a systematic and comprehensive whole. [4] The theoretical framework of the table is based on three partial characterizations of argument, namely (1) as first-order or second-order arguments ; (2) as predicate or subject arguments ; and (3) as a specific combination of types of statements. The superposition of these three partial characterizations yields a factorial typology of argument that can be used in order to develop tools for analyzing, evaluating, and producing argumentative discourse.

Every single type of argument described in the PTA consist of exactly one premise and one conclusion, both of which are expressed by means of a statement containing a subject and a predicate. Closely following logical conventions, subjects are indicated with letters a, b, etc., predicates with letters X, Y, etc. (predicate $\top$ having the fixed meaning 'true'), and complete propositions with letters p, q, etc. The identification of the type of argument takes place by following the so-called Argument Type Identification Procedure (ATIP), which is a heuristic device that helps the analyst

2. The concept of valency was introduced by Tesnière [12] within the framework of Structural Syntax. Gobbo and Benini [7] clarify the relation between CxAdGrams and that framework.

3. Please note that if the adposition of the second actant of 'to play' is filled by the preposition 'with', the overall semantics slightly changes. Thus, in CxAdGrams, 'to play' and 'to play with' can be treated as different verbs, if it is convenient. This apparently ad hoc treatment depends on English where a preposition may be used to modify the meaning of the verb, e.g., compare 'to get', 'to get out', 'to get off'. Such decisions lie beyond the scope of the present paper.

4. The present explanation of the PTA draws on [13, 14, 15, 17]. For regularly updated online information on related research projects, analyses of concrete examples, and downloads of relevant papers, see periodic-table-of-arguments.org.

to determine the 'argument form', a notion that comprises the first two partial characteristics mentioned above, as well as the 'argument substance', a notion that covers the third partial characteristic - see [16].

The theoretical framework of the PTA distinguishes between four basic argument forms : first-order predicate arguments, first-order subject arguments, second-order subject arguments, and second-order predicate arguments. In the visual representation of the table, these forms correspond to four different quadrants, which are indicated with Greek letters α, β, γ, and δ respectively. In Table 2, for each quadrant (Q) we list the corresponding argument form and provide a concrete example.

Q	*argument form*	*example*
α	a is X, because a is Y	The suspect (a) was driving fast (X), because he (a) left a long trace of rubber on the road (Y)
β	a is X, because b is X	Cycling on the grass (a) is forbidden (X), because walking on the grass (b) is forbidden (X)
γ	q is $\top$, because r is $\top$	He must have gone to the pub (q), because the interview was cancelled (r)
δ	q is $\top$, because q is Z	We only use 10% of our brain (q), because that (q) was said by Einstein (Z)

TABLE 2 – Argument forms and examples in the PTA

The argument types situated within each of the quadrants are further differentiated on the basis of a determination of the argument substance, i.e., the specific combination of types of statements. For this purpose, the PTA makes use of a tripartite typology consisting of statements of fact (F), statements of value (V), and statements of policy (P). Given these possibilities, the conclusion and premise of the argument substantiate one of the following nine combinations of types of statements : PP, PV, PF, VP, VV, VF, FP, FV, FF. The argument *The government should invest in jobs, because this will lead to economic growth*, for instance, can be characterized as a PF argument since its combines a statement of policy (P) in its conclusion with a statement of fact (F) in its premise.

When taken together, the three partial characterizations constitute a theoretical framework that allows for 2 x 2 x 9 = 36 different argument types. *The suspect was driving fast, because he left a long trace of rubber on the road*, for instance, can be identified as a first-order predicate argument that combines a statement of fact with another statement of fact. The systematic type indicator of this argument is therefore '1 pre FF'.

A final differentiation relates this systematic way of characterizing arguments to the traditional names that can be found in the existing dialectical and rhetorical classifications of argument. Each of the 36 argument types hosts an indefinite number of 'isotopes', which share the same characteristics, but differ as to the linguistic formulation of the connection between premise and conclusion. The key term of this formulation provides the name of the isotope. *The suspect was driving fast, because he left a long trace of rubber on the road*, to use the same example, can be identified as an 'argument from effect', since *leaving a long trace of rubber on the road* is an 'effect' of *driving fast*.

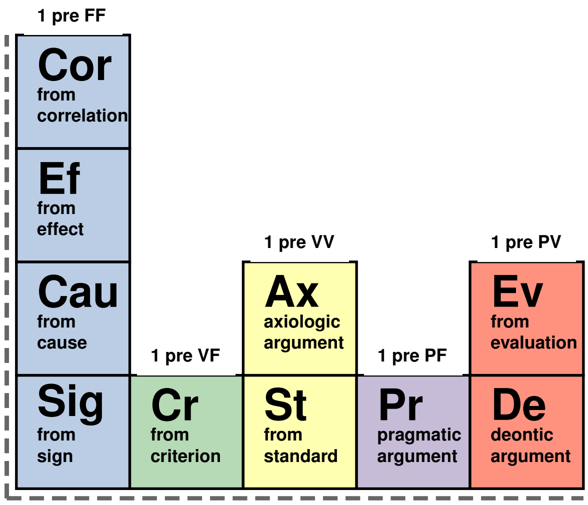

FIGURE 2 – The Alpha Quadrant of the PTA

The various types of argument are visually represented in the table in such a way that it becomes immediately clear what their characteristics are. The placement of an argument type within a particular quadrant provides information about its argument form – see Table 2. Within each of the quadrants, the horizontal placement of the argument types reflects the argument substance as expressed by the specific combination of types of statements – FF, VF, VV, etc. And finally, within every column, the vertical placement depends on the istope, thus reflecting the linguistic formulation of the relation between the premise and the conclusion.

In Figure 2, we showcase the Alpha Quadrant of the current version of the PTA (for the full picture, see its official website periodic-table-of-arguments.org). The argument from effect just mentioned is represented with the symbol 'Ef' and can be found in the leftmost column of this quadrant. It is accompanied by its isotopes, the 'argument from sign', the 'argument from cause', and the 'argument from correlation', which are all first-order predicate arguments based on different relationships between facts.

In the next section we demonstrate how the theoretical framework of the PTA can be used for enriching the linguistic adtrees generated by CxAdGrams with pragmatic information regarding the type of argument.

4 Building argumentative adpositional trees

While CxAdGrams is primarily built for expressing morphology and syntax, the adtrees generated by its theoretical framework are also suitable for expressing pragmatics. [5] In this section, we explain how to combine this formalism with that of the PTA and demonstrate how to insert pragmatic information concerning the characteristics of an argument into a linguistic adtree, thereby transforming it into a 'argumentative adtree' (or 'arg-adtree').

In order to build an arg-adtree, the analyst starts with constructing the two linguistic adtrees that represent the statements that function as the conclusion and the premise of the argument. [6]

Then, step by step, pragmatic information concerning the statements is added to the respective linguistic adtrees. This information includes their argumentative function as a conclusion or a premise. Conclusions are indicated by a sigma (σ), standing for the Greek equivalent $\sigma\upsilon\mu\pi\epsilon\rho\alpha\sigma\mu\alpha$ (sumpérasma), and premises by a pi (π), standing for $\pi\rho o\theta\alpha\sigma\iota\sigma$ (prótasis). It also includes the type of statement (P, V, or F) they substantiate, which is added under the function indicator.

Next, the two adtrees are conjoined, placing the conclusion on the right and the premise on the left. Depending on the order of presentation in the actual discourse, which can be progressive (premise, *therefore* conclusion) or retrogressive (conclusion, *because* premise), the arrow under the topmost hook points to the left or the right.

[7] Under this arrow, the analyst places pragmatic information about the type of argument, which includes the argument form (indicated by a Greek letter representing the corresponding quadrant) and its substance, the combination of types of statements (FF, VF, PF, etc.) Finally, the subjects and predicates of the statements are indicated by the same letters as the ones in use within the theoretical framework of the PTA (*a*, *b*, etc. for subjects and *X*, *Y*, etc. for predicates).

We will illustrate this process by constructing the arg-adtree of the example of the argument from effect mentioned in the previous section. This argument, *The suspect was driving fast, because he left a long trace of rubber on the road*, has been identified as a first-order predicate argument that supports a statement of fact with another statement of fact ('1 pre FF').

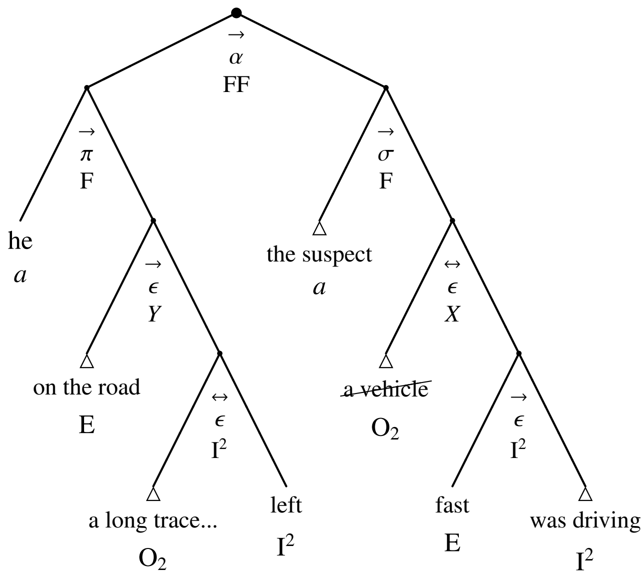

FIGURE 3 – The arg-adtree of the example

As pictured in Figure 3, the linguistic adtrees of the two statements that function as the premise and the conclusion of the argument have been conjoined and infused with pragmatic information so as to create the argumentative adtree. Under the topmost hook, information has been placed about the order of presentiation in the discourse (retrogressive, represented by →) as well as about the type of argument (1 pre FF, abbreviated as α FF). On the subsequent levels, apart from the linguistic information derived from the theoretical framework of CxAdGrams, pragmatic information is given regarding the argumentative function of the statements (conclusion, indicated by σ, and premise, indicated by π) as well as the type of statement (both are statements of fact, indicated by F).

Regarding the placement of the subject and predicate of the statements, the building of an argumentative adtree involves specific transformations of the linguistic adtrees. [8] While the position of the predicate does not change during this transformation process, in the argumentative adtree the subject is emphasized — see Figure 3. In this case, *he*, the first actant (O_1), functions as the subject of the premise (*a*) and is therefore put in evidence as the topmost left branch of the tree. The remaining linguistic material automatically becomes part of the predicate of the premise (*Y*).

5. Gobbo and Benini [6, ch. 6] have already provided pragmatic adtrees, representing the relation between illocutionary and locutionary acts as described in [11].

6. For an explanation of how to construct such linguistic adtrees, see [8]. The graphical aspect of an adtree is adapted for human readibility – or machine readability, in the case of linearisation. Such aspect follows some aesthetic and typographical – or coding – conventions, but what matters is how the various pieces are linked together.

7. Regarding the order of presentation of a premise and a conclusion, van Eemeren and Snoeck Henkemans [5, 33] distinguish between a progressive and a retrogressive mode. The former presents first the premise (π) and then the conclusion (σ), connecting them by means of the conjunction *so*, *therefore*, or another one with a similar function. Since there is a multitude of equivalent expressions, in argumentative adtrees of reconstructed arguments the progressive mode is represented only formally, by a left arrow (←). The retrogressive mode starts from the conclusion (σ) and then arrives at the premise (π), connecting them with a conjunction such as *because*, *since*, etc. This mode is represented in argumentative adtrees by a right arrow (→).

8. These transformations are formally justified by the so-called conjugate construction in the formal model of CxAdGrams – see [6, 211].

The reader is invited to note the corresponding tabularized form of this arg-adtree in Figure 4, which is useful for constructing a treebank of arg-adtrees suitable for natural language processing.

$\alpha_{FF}^{\rightarrow}($

  $\pi_{F}^{\rightarrow}($

    he_{a},

      $\epsilon_{X}^{\rightarrow}((\text{on the road})_{E}$

      ,

      $\epsilon_{X}^{\leftrightarrow}((\text{a long trace}\ldots)_{O_2}$

    $\text{left}_{I^2})))$,

  $\sigma_{F}^{\rightarrow}($

  ...

FIGURE 4 – Tabularization of the arg-adtree

The representation of the argument in an adtree reveals that there is a significant difference between the linguistic and the argumentative analysis : what is peripheral from a linguistic point of view, can be central from an argumentative perspective. In particular, the circumstantial *fast* (E) of the premise (π) is linguistically merely a decoration of the ruling verbal form *was driving* (I^2). In the analysis of the persuasive force of the argument, however, it plays a central role : if the suspect weren't driving *fast* (E), he could have never left a *long* trace (O_2) on the road.

As illustrated by this example, argumentative adtrees are powerful tools that enable the analyst to make explicit where the pragmatic force is placed within the linguistic material. In this way, the representation of the argument prepares the ground for its evaluation, indicating where to find the crucial 'points of attack'.

5 Conclusion

In this paper we demonstrated how the linguistic representation framework of *Constructive Adpositional Grammars* (CxAdGrams) and the argument classification framework of the *Periodic Table of Arguments* (PTA) can be combined so as to represent arguments in natural language. The method we developed for this purpose centers around the notion of 'argumentative adpositional tree' (or 'arg-adtree'). By providing an explanation of how to build an arg-adtree of an argument expressed in natural language, we made it clear how to represent not only the linguistic features of the statements involved, but also how to include pragmatic information about the overall structure of the argument, its type, and the argumentative function of its constituents. The method permits the analyst to operate on the level of the individual words and to freely choose the level of linguistic detail to be shown in the representation.

It is our aim to extend this combined approach, which we have named *Adpositional Argumentation* (AdArg), to the analysis of all the types of argument distinguished in the PTA. We will then apply it to concatenations of arguments, thereby providing a complete analysis of an argumentative text. In fact, since CxAdGrams provide a way to represent punctuation as conjunctions between sentences, they permit to represent a whole text in the terms of a single, comprehensive adtree.

In general, AdArg is aimed at fulfilling the need for a method for representing argumentative discourse that enables a formalization of the rich and detailed insights developed in the fields of Argumentation Theory and Rhetoric. Such a formalization, so we believe, is of use for researchers in the fields of Artificial Intelligence and Computational Argumentation, since it enables the development of tools that automatize to some extent the analysis and evaluation of argumentative discourse.

Regarding the possible application of the method presented in this paper, we think that it could be implemented in a computational model under a form of a tool. Such a tool would assist the analyst in making decisions regarding what linguistic and pragmatic information to include in specific reconstructions of argumentative discourse. Whereas the state-of-the-art in Computational Argumentation has automatized the extraction of complete propositions and their relations, our approach can be used to develop tools for a more fine-grained computer-assisted analysis of argumentative texts. If the linguistic and pragmatic information included in our framework is combined with example-based data extracted from past analyses, it would become possible to partially automatize the whole procedure using Artificial Intelligence techniques.

Acknowledgements

This paper is an amended merger of two earlier papers in which we have explained our method for building argumentative adpositional adtrees for different readerships [8, 9]. The authors thank Marco Benini for his thorough reading of the manuscript, and in particular for checking the parts concerning constructive mathematics.

Références

[1] Adjukiewicz, K.: *Die syntaxische Konnexität*. Studia Philosophica, 1 :1–27, 1935.

[2] Bridges, D. et F. Richman: *Varieties of Constructive Mathematics*. Cambridge University Press, 1987.

[3] Church, A.: *A formulation of the simple theory of types*. Journal of Symbolic Logic, 5:56–58, 1940.

[4] Eemeren, F. H. van, B. J. Garssen, E. C. W. Krabbe, A. F. Snoeck Henkemans, H. B. Verheij et J. H. M. Wagemans: *Handbook of Argumentation Theory*. Springer, 2014.

[5] Eemeren, F. H. van et A. F. Snoeck Henkemans: *Argumentation. Analysis and evaluation*. Routledge, 2^{e} édition, 2016.

[6] Gobbo, F. et M. Benini: *Constructive Adpositional Grammars. Foundations of Constructive Linguistics*. Cambridge Scholars Publishing, 2011.

[7] Gobbo, F. et M. Benini: *Dependency and valency. from structural syntax to constructive adpositional grammars*. Dans *Computational Dependency Theory*, pages 113–135, 2013.

[8] Gobbo, F. et J. H. M. Wagemans: *Building argumentative adpositional trees: Towards a high precision method for reconstructing arguments in natural language*. Dans *Proceedings of the 9th Conference of ISSA*, pages 408–420, 2019.

[9] Gobbo, F. et J. H. M. Wagemans: *A method for reconstructing first-order arguments in natural language*. Dans *Proceedings of AIXIA 2018*, pages 27–23, 2019.

[10] Modgil, S., K. Budzynska et J. Lawrence: *Computational models of argument*. Dans *Proceedings of COMMA 2018*, 2019.

[11] Searle, J. R.: *Making the social world. The structure of human civilization*. Oxford University Press, 2010.

[12] Tesnière, L.: *Éléments de Syntaxe Structurale*. Klincksieck, 1959.

[13] Wagemans, J. H. M.: *Constructing a periodic table of arguments*. Dans *Argumentation, Objectivity, and Bias: Proceedings of the 11th International Conference of OSSA*, 2016.

[14] Wagemans, J. H. M.: *Analogy, similarity, and the periodic table of arguments*. Studies in Logic, Grammar and Rhetoric, 55:63–75, 2018.

[15] Wagemans, J. H. M.: *Assertoric syllogistic and the periodic table of arguments*. Dans *Argumentation and Inference: Proceedings of the 2nd ECA*, 2018.

[16] Wagemans, J. H. M.: *Argument type identification procedure (atip)*. Published online, jan 2019. `periodic-table-of-arguments.org/argument-type-identification-procedure`.

[17] Wagemans, J. H. M.: *Four basic argument forms*. Research in Language, 17:57–69, 2019.