

Representing and Reasoning about Arguments Mined from Texts and Dialogues

Leila Amgoud¹, Philippe Besnard¹, and Anthony Hunter²

¹ CNRS, IRIT, Université de Toulouse, Toulouse, France

² University College London, London, U.K.

Abstract. This paper presents a target language for representing arguments mined from natural language. The key features are the connection between possible reasons and possible claims and recursive embedding of such connections. Given a base of these arguments and counterarguments mined from texts or dialogues, we want be able combine them, deconstruct them, and to analyse them (for instance to check whether the set is inconsistent). To address these needs, we propose a formal language for representing reasons and claims, and a framework for inferencing with the arguments and counterarguments in this formal language.

1 Introduction

There is growing interest in the computational linguistics community in identifying arguments and relations between them in natural language (see for example [1–3]). Consider also the First ACL Workshop on Argument Mining held in 2014 [4], and the IBM Debating Technologies being developed by IBM Research for extracting arguments from sources such as Wikipedia [5, 6].

An interesting challenge that is arising from attempts to mine arguments (seen as *reasons* about *claims*) is the choice of target formalism for representing the extracted arguments. In computational models of argument, abstract argumentation (as proposed by Dung [7]) and logical or structured argumentation (as proposed in [8–11]) are the two key options. Neither is ideal as a target formalism as we outline below.

Abstract argumentation: Each argument is atomic. Thus, as a target language, it is quite weak since there is no formal differentiation of reason and claim. A ramification of this is that there can be no recursive embedding (e.g. reasons for the claim “*x is a reason for claim y*” cannot be represented in abstract argumentation). Also, it does not allow for Boolean combinations of reasons like “*x or y is a reason for z*”. In addition, it does not differentiate between i) “*x being a reason for not claiming y*” and ii) “*x not being a reason for claiming y*”. Introducing a support relation partly addresses some shortcomings of abstract argumentation, but not the ones mentioned above. Note that the notion of a supporting argument is not necessarily the same as the notion of a reason.

Logical argumentation: Each argument is a set of formulae entailing a claim. As a target language, its expressiveness does not fit with the above needs. Firstly, all the logical formulae for the premises and claims need to be identified. It is currently not feasible to aim for translating argumentation in text and dialogues into logical formulae such that all the premises are represented explicitly and the claim follows logically from

the premises. One great challenge for this is that most arguments in text and dialogues are *enthymemes*. That is, some or all of the premises are *implicit*, and even the claim may be implicit. For more on computational modelling of enthymemes, see [12]. Logical argumentation is also insufficient because it does not capture embedded relationships (like “*(x is not a reason for y) is a reason for z*”) nor does it explicitly capture “*x is not a reason for claim y*” or “*x is a reason for not claiming y*”. So, as a target language, logical argumentation captures details that cannot be extracted in the short term by argument mining and it lacks some features potentially valuable for argument mining.

Our proposal here is partly motivated by the need for a better formalism for representing argumentation coming from natural language. There is a second motivation. Given a set of arguments and counterarguments represented in our target language, we can infer further arguments and counterarguments. This is not possible with abstract argument since it has no inference machinery over argument graphs. This is also not possible directly with logical argumentation. Given logical arguments A_1 and A_2 , we cannot infer a new logical argument A_3 (other than by extracting the support in each argument and using that as a knowledge base for generating further logical arguments).

To address these, we propose a formal language for representing reasons and claims, as well as a framework for inferring with arguments and counterarguments in this language. The implied arguments will be deconstructions, e.g. obtained by flattening the recursive structure in an argument, and constructions obtained by combining arguments. This framework is flexible in that different choices for inference rules can be made.

2 Formal syntax

We now present our formalism for representing arguments, inspired by Apothéloz [13]. The formalism is built upon a propositional language \mathbb{L} with the connectives \neg , \vee , \wedge , \rightarrow , \leftrightarrow . There are also two operators $\mathcal{R}(\cdot)$ and $\mathcal{C}(\cdot)$ and an additional negation $-$. Thus, two negation operators are needed: \neg for denying propositional formulas ($\neg x$ denotes that x is false), and $-$ for denying $\mathcal{R}(\cdot)$ and $\mathcal{C}(\cdot)$. Please note that $\neg\neg x$ is identified with x and $- - \mathcal{R}(\cdot)$ is identified with $\mathcal{R}(\cdot)$ (similarly, $- - \mathcal{C}(\cdot)$ is identified with $\mathcal{C}(\cdot)$).

Definition 1 (Formula). *The set of formulas L_A is the smallest set such that a formula is of the form $(-)\mathcal{R}(y) : (-)\mathcal{C}(x)$ where each of x and y is either a formula of \mathbb{L} or is itself a formula.*

Each formula is either an argument or a rejection of an argument, to be defined next.

Definition 2 (Argument). *An argument is a formula of L_A of the form $\mathcal{R}(y) : (-)\mathcal{C}(x)$.*

An *argument* is a reason for concluding a claim. It has two main parts: *premises* (the reason) and a *conclusion*. The functions \mathcal{R} and \mathcal{C} respectively play the roles of *giving reason* and *concluding*. Indeed, an argument is interpreted as follows: its conclusion holds *because* it follows, according to a given notion, from the premises. The notion refers to the nature of the link between them (e.g., the premises imply the conclusion), formally identified by the colon in the definition. However, the contents may be true while the functions do not hold and vice versa. The intuitive reading is as follows:

$$\begin{aligned} \mathcal{R}(y) : \mathcal{C}(x) &\text{ means that “}y\text{ is a reason for concluding }x\text{”} \\ \mathcal{R}(y) : -\mathcal{C}(x) &\text{ means that “}y\text{ is a reason for not concluding }x\text{”} \end{aligned}$$

The first kind of expression captures two forms of argument about a proposition x : One in which x is supported ($\mathcal{R}(y) : \mathcal{C}(x)$) and one in which its negation is supported ($\mathcal{R}(y) : \mathcal{C}(\neg x)$). As to the expression $\mathcal{R}(y) : \neg \mathcal{C}(x)$, it encompasses two cases:

- the case where y is a reason for concluding $\neg x$;
- the case that y does not support $\neg x$ but still does not support x either.

Example 1 In the dialogue next, Mary's argument supports $\neg fe$, a claim opposing Paul's but John's argument is only meant to stop concluding fe without committing to $\neg fe$.

Paul: Carl will fail his exams (fe). He did not work hard ($\neg wh$). $\mathcal{R}(\neg wh) : \mathcal{C}(fe)$
 Mary: No, he will not fail. The exams will be easy (ee). $\mathcal{R}(ee) : \mathcal{C}(\neg fe)$
 John: Carl is very smart! (sm). $\mathcal{R}(sm) : \neg \mathcal{C}(fe)$

Unlike existing definitions of argument where a conclusion x follows from premises y using a notion of *derivation* [8], Definition 2 does not make the link explicit. We aim at a general definition that allows us to represent any argument in text or dialogue, e.g. enthymemes [12], without judging whether it is a good argument. Also, reasons can be hypothetical (assumed for the purpose of the argument). And our approach is meant to capture links of whatever nature including non-deductive links as in the causal argument

Ice on its wings (iw) could cause a plane to crash (pc). $\mathcal{R}(iw) : \mathcal{C}(pc)$

or abductive links as in the following argument:

Tim and Jack have recently had a row that ended their friendship (x). Clara just saw them jogging together (y). Thus, they are friends again (z). $\mathcal{R}(x \wedge y) : \mathcal{C}(z)$

So far, the negation operator “ \neg ” has been used to deny the concluding function. Now, the function of *giving reason* can be denied as well by placing “ \neg ” in front of \mathcal{R} . What is denied in this case is not the premises but rather the idea that the premises justify the conclusion of the argument. Such a form is called *rejection* of argument.

Definition 3 (Rejection). A rejection of an argument is a formula of L_A of the form $\neg \mathcal{R}(y) : (-)\mathcal{C}(x)$.

The intuitive meaning for these formal expressions is as follows:

$\neg \mathcal{R}(y) : \mathcal{C}(x)$ means that “ y is not a reason for concluding x ”
 $\neg \mathcal{R}(y) : \neg \mathcal{C}(x)$ means that “ y is not a reason for not concluding x ”

Example 2 Consider the following dialogue.

Eric: The fact that Carl is smart is not a reason to stop concluding that he will fail his exams. $\neg \mathcal{R}(sm) : \neg \mathcal{C}(fe)$
 John: Anyway, the fact that Carl did not work hard is not a reason to conclude that he will fail his exams. $\neg \mathcal{R}(\neg wh) : \mathcal{C}(fe)$
 Ann: Being stressed is the reason that Carl will fail his exams, hence it is not the fact that he did not work hard (st). $\mathcal{R}(\mathcal{R}(st) : \mathcal{C}(fe)) : \mathcal{C}(\neg \mathcal{R}(\neg wh) : \mathcal{C}(fe))$
 Sara: He is not stressed at all. $\mathcal{R}(\neg st) : \mathcal{C}(\neg \mathcal{R}(st) : \mathcal{C}(fe))$

There can be many reasons for rejecting an argument $\mathcal{R}(y) : \mathcal{C}(x)$ as illustrated next:

1. y is true but is *irrelevant* to x .
The fact that the weather is cloudy (wcl) is no reason to conclude that Carl will fail his exams. $-\mathcal{R}(wcl) : \mathcal{C}(fe)$
2. y is true, relevant to x but *not sufficient* to explain x .
Paul: You should buy the same car as mine (bc). It's fast (f). $\mathcal{R}(f) : \mathcal{C}(bc)$
John: If it's affordable! (a). $\mathcal{R}(\mathcal{R}(f \wedge a) : \mathcal{C}(bc)) : \mathcal{C}(-\mathcal{R}(f) : \mathcal{C}(bc))$
3. y is false, thus y cannot be a reason for x .
 Sara's argument is an example.
4. y and x are both true but y is not the reason for x (there is a better reason for x).
 Ann's argument is an example.

Forms 1-3 of rejection amount to blocking the conclusion of the targeted argument. They give rise to arguments whose reason justifies rejecting the targeted argument and whose conclusion inhibits the conclusion of the targeted argument: Eric's rejection leads to $\mathcal{R}(-\mathcal{R}(sm) : -\mathcal{C}(fe)) : \mathcal{C}(fe)$ and John's to $\mathcal{R}(-\mathcal{R}(\neg wh) : \mathcal{C}(fe)) : -\mathcal{C}(fe)$. As to form 4, the reason and conclusion of the targeted argument are acknowledged. Such a rejection is less of a counterargument than a better argument for the conclusion.

No other logic-based approach to modelling argumentation provides a language for expressing rejection of arguments in the object language. This gives a more appropriate encoding of situations and allows us to differentiate between say $-\mathcal{R}(cr) : \mathcal{C}(bc)$ and $\mathcal{R}(cr) : -\mathcal{C}(bc)$ (let cr denote "The car is red" and bc denote "We should buy the car").

- $-\mathcal{R}(cr) : \mathcal{C}(bc)$ could represent a rejection of the argument $\mathcal{R}(cr) : \mathcal{C}(bc)$ as we need to consider more than the colour of the car when choosing whether to buy it.
- $\mathcal{R}(cr) : -\mathcal{C}(bc)$ could represent a rejection of $\mathcal{R}(cr) : \mathcal{C}(bc)$ because we do not like the colour red for a car.

Even if we identify the rejection $-\mathcal{R}(cr) : \mathcal{C}(bc)$, it is possible that we could identify another argument for buying the car using other criteria such as $\mathcal{R}(ec \wedge sp) : \mathcal{C}(bc)$ where ec denotes "The car is economical" and sp denotes "The car is spacious".

3 Representing mined arguments

Our approach first aims at representing mined arguments. We want to use our language to represent arguments as they arise in natural language texts and dialogues. We believe that central to handling texts and dialogues is the need to provide support for nested arguments and rejections. To illustrate some of the richness of our approach, we give in Table 1 various forms (F_1 - F_{12}) of arguments and rejections allowed by our definitions (x, y, z, t are propositional formulas to simplify matters). The table is not exhaustive.

In our approach, we are not identifying what constitutes a good argument (or a good rejection of an argument). Rather, we are providing a representation of arguments (and rejections thereof). If an argument or rejection occurs in a text or dialogue, then we want it to be mined, and we want to be able to represent it in our language.

A list of arguments below shows that all the forms F_i can be used as a target for natural language. It indicates how to use our language, rather than suggesting that there is a canonical translation of text in to the formal target language. As with translating a natural language sentence in to any logic, there is some subjectivity in how a sentence is exactly translated, depending on various factors.

Table 1

<i>Basic arguments</i>	F_1	$\mathcal{R}(y) : \mathcal{C}(x)$
	F_2	$\mathcal{R}(y) : \mathcal{C}(\neg x)$
	F_3	$\mathcal{R}(y) : \neg \mathcal{C}(x)$
<i>Single-embedding meta-arguments (in reason)</i>	F_4	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \mathcal{C}(x)$
	F_5	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \mathcal{C}(\neg x)$
	F_6	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \neg \mathcal{C}(x)$
<i>Single-embedding meta-arguments (in conclusion)</i>	F_7	$\mathcal{R}(y) : \mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))$
	F_8	$\mathcal{R}(y) : \mathcal{C}(\neg \mathcal{R}(z) : \mathcal{C}(x))$
	F_9	$\mathcal{R}(y) : \neg \mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))$
<i>Double-embedding meta-arguments</i>	F_{10}	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \mathcal{C}(\mathcal{R}(t) : \mathcal{C}(x))$
	F_{11}	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \mathcal{C}(\neg \mathcal{R}(t) : \mathcal{C}(x))$
	F_{12}	$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \neg \mathcal{C}(\mathcal{R}(t) : \mathcal{C}(x))$

- F_1 : Tweety can fly (f). It is a bird (b). $\mathcal{R}(b) : \mathcal{C}(f)$
- F_2 : Tweety cannot fly. It is a penguin (p). $\mathcal{R}(p) : \mathcal{C}(\neg f)$
- F_3 : Carl is smart. Thus, it is not possible to conclude that he will fail his exams. $\mathcal{R}(sm) : \neg \mathcal{C}(fe)$
- F_4 : That Tweety can fly because it is a bird, is a reason to conclude that Tweety has wings (w). $\mathcal{R}(\mathcal{R}(b) : \mathcal{C}(f)) : \mathcal{C}(w)$
- F_5 : That Carl will fail his exams because he did not work hard is a reason to conclude that he is not so smart. $\mathcal{R}(\mathcal{R}(\neg wh) : \mathcal{C}(fe)) : \mathcal{C}(\neg sm)$
- F_6 : Paul's car is in his job parking lot (pr) because it is broken (br), hence we cannot conclude that Paul is in his office (of). $\mathcal{R}(\mathcal{R}(br) : \mathcal{C}(pr)) : \neg \mathcal{C}(of)$
- F_7 : The weather is sunny (su). Thus, rain (ra) will lead to rainbow (rb). $\mathcal{R}(su) : \mathcal{C}(\mathcal{R}(ra) : \mathcal{C}(rb))$
- F_8 : The fact that Tweety is a penguin is a reason to conclude that being a bird is not a sufficient reason for Tweety being able to fly. $\mathcal{R}(p) : \mathcal{C}(\neg \mathcal{R}(b) : \mathcal{C}(f))$
- F_9 : The fact that all European countries have a strong economy (se) is a reason for not concluding that an economic crisis (ec) in Germany is a reason for a declining value of the euro (de). $\mathcal{R}(se) : \neg \mathcal{C}(\mathcal{R}(ec) : \mathcal{C}(de))$
- F_{10} : CFCs (cfc) cause damage to the ozone layer of the atmosphere (do). Man-made pollution (mp) causes global warming (gw). $\mathcal{R}(\mathcal{R}(cfc) : \mathcal{C}(do)) : \mathcal{C}(\mathcal{R}(mp) : \mathcal{C}(gw))$
- F_{11} : Stress is the reason that Carl will fail his exams, hence it is not the fact that he did not work hard (st). $\mathcal{R}(\mathcal{R}(st) : \mathcal{C}(fe)) : \mathcal{C}(\neg \mathcal{R}(\neg wh) : \mathcal{C}(fe))$
- F_{12} : The object looks red (lr). It is illuminated by red light (il). Thus, we cannot conclude that looking red implies the object being indeed red (re). $\mathcal{R}(\mathcal{R}(il) : \mathcal{C}(lr)) : \neg \mathcal{C}(\mathcal{R}(lr) : \mathcal{C}(re))$

Examples in F_1 - F_{12} illustrate that the outer reason and claim can be potentially identified using argument mining techniques, such as based on sentiment analysis techniques (e.g. [14]), text entailment techniques (e.g. [1]), or directly via machine learning techniques (e.g. [6]), and then by recursion, the inner reasons and claims can be identified by argument mining techniques. Thus, the nested structure appears better suited as a target language for arguments as they arise in natural language dialogues and texts.

4 Reasoning systems

Our approach also aims at reasoning with mined arguments and rejections. We want to find what arguments and rejections follow from them. Note that deriving an argument α does *not* mean that α is accepted. Instead, inferring α means that the argument(s) used while deriving α cannot be held without α also being held. Inference captures *commitment* between arguments. Hence, if a foolish argument is used as a premise then a foolish α may result: If an agent holds a foolish argument, he henceforth commits to some other foolish arguments.

Our approach is to treat a set of arguments and rejections as a set of formulae from which we apply a reasoning system. A consequence operator \Vdash is the least closure of a set of *inference rules* extended with one *meta-rule*. Different reasoning systems can be defined by adopting different choices for the set of inference rules. In this paper, we only have space to introduce one reasoning system (in Section 5), but we will also indicate how alternative reasoning systems can be identified.

For the sake of simplicity, the rules are introduced using propositional formulas. However, they *all* hold in the general case. So, the variables x, y, z can be instantiated with propositional formulae, arguments, and rejections of arguments.

Let us introduce the meta-rule. In keeping with the meaning of $-\mathcal{R}(y) : \mathcal{C}(x)$, i.e., y is not a reason for x , which is the negation of the meaning of $\mathcal{R}(y) : \mathcal{C}(x)$, i.e., y is a reason for x , the *meta-rule* expresses that we can reverse any inference rule of the form

$$\frac{\mathcal{R}(y) : \Phi}{-\mathcal{R}(y) : \Psi} \quad \text{into} \quad \frac{\mathcal{R}(y) : \Psi}{-\mathcal{R}(y) : \Phi}$$

Of course, the same reversing process takes place whenever “-” occurs in front of a leftmost “ \mathcal{R} ” so that, in the general case, an inference rule ¹ where $i, j \in \{0, 1\}$

$$\frac{-^{(i)}\mathcal{R}(y) : \Phi}{-^{(j)}\mathcal{R}(y) : \Psi} \quad \text{can be reversed into} \quad \frac{-^{(1-j)}\mathcal{R}(y) : \Psi}{-^{(1-i)}\mathcal{R}(y) : \Phi}$$

Now we turn to introducing the inference rules. As a start, consistency of arguments’ reasons is a source of inferences (where x is a formula in \mathbb{L}).

$$\frac{\mathcal{R}(y) : \mathcal{C}(x)}{-\mathcal{R}(y) : -\mathcal{C}(x)} \quad \frac{\mathcal{R}(y) : \mathcal{C}(x)}{\mathcal{R}(y) : -\mathcal{C}(\neg x)} \quad \text{(Consistency)}$$

The leftmost rule means that if y is a reason for x then y is not a reason to doubt x . The rightmost rule means that if y is a reason for x then it is also a reason to doubt $\neg x$.

Proposition 1 *The inference rules below are derived from (Consistency) and the meta-rule (where x is a formula in \mathbb{L} in the first, third and fourth inference rules).*

$$\frac{\mathcal{R}(y) : \mathcal{C}(x)}{-\mathcal{R}(y) : \mathcal{C}(\neg x)} \quad \frac{\mathcal{R}(y) : -\mathcal{C}(x)}{-\mathcal{R}(y) : \mathcal{C}(x)} \quad \frac{\mathcal{R}(y) : \mathcal{C}(\neg x)}{\mathcal{R}(y) : -\mathcal{C}(x)} \quad \frac{\mathcal{R}(y) : \mathcal{C}(x)}{-\mathcal{R}(y) : \mathcal{C}(x)}$$

Any reasoning system for our language is to include the meta-rule and the derived rules. In Section 5, we consider specific inference rules for an example of a reasoning system.

¹ $-^{(1)}$ denotes a single occurrence of the hyphen and $-^{(0)}$ the absence of it.

5 Indicative reasoning

Indicative reasoning is an example of a reasoning system for our language. We give for this system some simple inference rules that can be used with the meta-rule presented in Section 4. We discuss some inference rules that are not part of the set of rules for this example of a reasoning system and give some properties of the consequence relation.

5.1 Inference rules

The first inference rule for indicative reasoning captures when reasons are interchangeable. This is when x is the reason for y and vice-versa. Hence the next inference rule.

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad \mathcal{R}(x) : \mathcal{C}(y) \quad \mathcal{R}(y) : \mathcal{C}(z)}{\mathcal{R}(x) : \mathcal{C}(z)} \quad \text{(Mutual Support)}$$

As an illustration, let x stand for “Paul and Mary are married to each other” and y for “Paul and Mary are in love with each other”.

The next rule gathers different reasons for the same conclusion within a single argument where y and z are formulas in \mathbb{L} .

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad \mathcal{R}(z) : \mathcal{C}(x)}{\mathcal{R}(y \vee z) : \mathcal{C}(x)} \quad \text{(Or)}$$

Cautious monotonicity means that the reason of an argument can be expanded with any premise it justifies. *Cut* expresses a form of minimality of the reason of an argument.

$$\begin{array}{cc} \text{(Cautious Monotonicity)} & \text{(Cut)} \\ \frac{\mathcal{R}(y) : \mathcal{C}(z) \quad \mathcal{R}(y) : \mathcal{C}(x)}{\mathcal{R}(y \wedge z) : \mathcal{C}(x)} & \frac{\mathcal{R}(y \wedge z) : \mathcal{C}(x) \quad \mathcal{R}(y) : \mathcal{C}(z)}{\mathcal{R}(y) : \mathcal{C}(x)} \end{array}$$

The two next rules concern nesting of $\mathcal{R}(\cdot)$ and $\mathcal{C}(\cdot)$. *Exportation* shows how to simplify meta-arguments (where y and z are formulas in \mathbb{L}) and *Permutation* shows that for some forms of meta-arguments, permutations of reasons are possible.

$$\begin{array}{cc} \text{(Exportation)} & \text{(Permutation)} \\ \frac{\mathcal{R}(y) : \mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))}{\mathcal{R}(y \wedge z) : \mathcal{C}(x)} & \frac{\mathcal{R}(y) : \mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))}{\mathcal{R}(z) : \mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x))} \end{array}$$

We show that the consequence operator \Vdash defined upon the introduced inference rules is consistent. Indeed, $\neg\mathcal{R}(y) : \mathcal{C}(x)$ cannot be schematically derived from $\mathcal{R}(y) : \mathcal{C}(x)$ and that $\neg\mathcal{R}(y) : \neg\mathcal{C}(x)$ cannot be schematically derived from $\mathcal{R}(y) : \neg\mathcal{C}(x)$. By the inference rules in Property 1, Property 2 actually expresses that neither $\mathcal{R}(y) : \mathcal{C}(\neg x)$ nor $\mathcal{R}(y) : \neg\mathcal{C}(x)$ can be schematically derived from $\mathcal{R}(y) : \mathcal{C}(x)$.

Proposition 2 *There is no $i, j \in \{0, 1\}$ s.t. the following is a derived inference rule.*

$$\frac{\neg^{(i)}\mathcal{R}(y) : \neg^{(j)}\mathcal{C}(x)}{\neg^{(1-i)}\mathcal{R}(y) : \neg^{(j)}\mathcal{C}(x)}$$

5.2 Non-Inference

We turn to inference rules that are excluded from our example of a reasoning system. We explain why they are excluded but some of these could be in an alternative system.

Proposition 3 *Neither the (Reflexivity) axiom, i.e., $\mathcal{R}(x) : \mathcal{C}(x)$ for all $x \in \mathbb{L}$, nor the following rules hold in indicative reasoning.*

<p>(Logical Consequence)</p> $\frac{y \models x}{\mathcal{R}(y) : \mathcal{C}(x)}$	<p>(Left Logical Equivalence)</p> $\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad \models y \leftrightarrow z}{\mathcal{R}(z) : \mathcal{C}(x)}$
<p>(Left Logical Consequence)</p> $\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad z \models y}{\mathcal{R}(z) : \mathcal{C}(x)}$	<p>(Right Logical Consequence)</p> $\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad x \models z}{\mathcal{R}(y) : \mathcal{C}(z)}$
<p>(Transitivity)</p> $\frac{\mathcal{R}(z) : \mathcal{C}(y) \quad \mathcal{R}(y) : \mathcal{C}(x)}{\mathcal{R}(z) : \mathcal{C}(x)}$	<p>(And)</p> $\frac{\mathcal{R}(y) : \mathcal{C}(x) \quad \mathcal{R}(y) : \mathcal{C}(z)}{\mathcal{R}(y) : \mathcal{C}(x \wedge z)}$

Reflexivity is omitted because it seems unlikely that in full generality x be a reason for x , e.g., not when the link is causality. Hence *Logical Consequence* is inhibited as well. Indeed, being a reason for x is far *more restrictive* than having x as logical consequence. Here is an illustration. By classical logic, taking x to stand for “if I am in London then I am in England” while taking y to be $\neg z$ where z stands for “if I am in England then I am in Paris” yields $\neg z \models x$. However, $\neg z$ falls short of being a reason for x .

We refrain from adopting *Right Logical Consequence* in indicative reasoning again on the grounds that being a reason for x is in general far more restrictive than having x as a logical consequence. Indeed, the *nature* of the link plays an important role. E.g., consider the causal argument: flu is a reason for your body temperature to be in the range $39^\circ\text{C}–41^\circ\text{C}$. The fact that being in the range $36^\circ\text{C}–41^\circ\text{C}$ is a logical consequence of being in the range $39^\circ\text{C}–41^\circ\text{C}$ does not make flu a reason for your body temperature to be in the range $36^\circ\text{C}–41^\circ\text{C}$ (it is the only possible range unless you are dead!).

And cannot be adopted either for indicative reasoning. Consider the case where y , x and z stand respectively for “Paul is standing in the middle of the road while a car is approaching”, “Paul should move forward”, and “Paul should move back”.

Transitivity does not hold either for indicative reasoning. Back to the F_6 example, the fact that Paul’s car is broken does not support the conclusion “Paul is in his office”.

In indicative reasoning, blocking a reason is different from blocking a conclusion:

(a) $\mathcal{R}(y) : \neg\mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))$ does not imply $\mathcal{R}(y) : \neg\mathcal{C}(x)$

(b) $\mathcal{R}(y) : \neg\mathcal{C}(x)$ does not imply $\mathcal{R}(y) : \neg\mathcal{C}(\mathcal{R}(z) : \mathcal{C}(x))$

Argument 1: *The fact that several European countries have a good economy (ge) is a reason for not concluding that an economic crisis (ec) in Spain is a reason for a declining value of the euro (de).*

Argument 1 has the form $\mathcal{R}(ge) : \neg\mathcal{C}(\mathcal{R}(ec) : \mathcal{C}(de))$ and illustrates case (a) because $\mathcal{R}(ge) : \neg\mathcal{C}(de)$ needs not hold since an economic crisis in Germany may lead to a declining value of the euro.

Argument 2: *The fact that Carl did not attend at all the course (ac) is a reason for him failing his exams.*

That Argument 2, formally captured as $\mathcal{R}(\neg ac) : \mathcal{C}(fe)$, is doubted on the grounds that Carl is smart can be written $\mathcal{R}(sm) : \neg\mathcal{C}(\mathcal{R}(\neg ac) : \mathcal{C}(fe))$. However, the latter argument needs not hold even in the presence of $\mathcal{R}(sm) : \neg\mathcal{C}(fe)$ (Carl being smart is a reason not to conclude him failing his exams) because $\neg ac$ is *more specific* than sm .

We insist that the indicative reasoning system we present here is only one option for a reasoning system with our approach. By giving this system, we also want to question the appropriateness of some simple inference rules that most notions of logic would adhere to. In particular, we suggest that reasoning with arguments needs not use the same inference rules as non-monotonic reasoning (cf e.g. Kraus *et al* [15]). For instance, we may choose to set up a system such as the indicative reasoning system, that fails reflexivity (i.e. we do not accept that statement x is automatically a reason for claim x).

5.3 Consequence relation

When \Vdash is the smallest inference relation obeying the rules from Section 5.1, reflexivity, monotonicity and cut hold, meaning that with the \Vdash consequence relation, manipulation of arguments by the inference rules is *well-founded*, in the logic tradition [16]. Indeed:

Proposition 4 *Let Δ be a set of (rejections of) arguments. Let α , and β be arguments.*

$\Delta \Vdash \alpha$ if $\alpha \in \Delta$	(Reflexivity)
$\Delta \cup \{\alpha\} \Vdash \beta$ if $\Delta \Vdash \beta$	(Monotonicity)
$\Delta \Vdash \beta$ if $\Delta \cup \{\alpha\} \Vdash \beta$ and $\Delta \Vdash \alpha$	(Cut)

Also, the \Vdash consequence relation is *paraconsistent* in the sense that it is not trivialized by contradiction (i.e., not all formulae of the language L_A follow from contradiction).

Proposition 5 $\{-^{(i)}\mathcal{R}(y) : -^{(j)}\mathcal{C}(x), -^{(1-i)}\mathcal{R}(y) : -^{(j)}\mathcal{C}(x)\} \not\vdash L_A$

\Vdash is monotonic but involves a non-monotonic operator in its object language: \mathcal{R} . “being a reason” is a non-monotonic relation $\vdash\sim$ as witnessed by transitivity failing (it can be that y is generally a reason for x although there are special cases where this fails). As an aside, please note that Mutual Support is a special instance of Transitivity. Anyway, the fact that $\vdash\sim$ is formalized as an operator in our formalism makes the non-monotonicity confined to failure of inferring $\mathcal{R}(y \wedge z) : \mathcal{C}(x)$ from $\mathcal{R}(y) : \mathcal{C}(x)$. There is no effect on the logic. The case is similar to conditional logics (they *are* monotonic) because an operator capturing a counterfactual conditional must be *non-monotonic*: e.g., “were I to scratch this match, it would ignite” denoted $y \rightarrow x$ may hold while “were I to scratch this match, that is wet, it would ignite” denoted $y \wedge z \rightarrow x$ fails to hold.

6 Conclusion

We propose a logic for representing and reasoning about arguments in a way that is just not possible with existing formalisms. We think that the formalism is a promising

target language for argument mining and that if we obtain arguments (and rejections of arguments) by argument mining, we can use our inference machinery to analyse them.

Our formalism captures arguments and rejections thereof, with definitions encompassing different roles of reasons (concluding and blocking statements), various forms of reasons (factual and hypothetical) and different kinds of links (deductive, abductive, inductive, . . .). Unlike existing computational models of argumentation where attack (support) between arguments is expressed by external relations on the set of arguments, any attack (support) can be expressed as an argument in our formalism.

Many well-known logics were first proposed as proof systems. For these logics, semantics were only obtained later. We have identified an alternative reasoning system (that preserves more of the classical inference rules such as reflexivity) with a three-valued semantics. An exciting challenge is to identify a semantics for indicative reasoning. Future research includes exploring the space of reasoning systems, semantics for them, and investigating Boolean combinations of arguments (and rejections thereof).

References

1. Cabrio, E., Villata, S.: Generating abstract arguments: a natural language approach. In: Proc. 4th C. Computational Models of Argument (COMMA'12), IOS Press (2012) 454–461
2. Florou, E., Konstantopoulos, S., Koukourikos, A., Karampiperis, P.: Argument extraction for supporting public policy formulation. In: Proc. 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, ACL (2013) 49–54
3. Peldszus, A., Stede, M.: From argument diagrams to argumentation mining in texts: a survey. *Int. Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 7(1) (2013) 1–31
4. Green, N., Ashley, K., Litman, D., Reed, C., Walker, V., eds.: Proceedings of the First ACL Workshop on Argumentation Mining. (2014)
5. Aharoni, E., Polnarov, A., Lavee, T., Hershovich, D., Levy, R., Rinott, R., Gutfreund, D., Slonim, N.: A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In: Proc. ACL Workshop on Argumentation Mining. (2014)
6. Levy, R., Bilu, Y., Hershovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: Proc. 25th Int. Conf. Computational Linguistics (COLING). (2014)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artific. Intelligence* 77 (1995) 321–357
8. Besnard, P., Hunter, A.: *Elements of Argumentation*. MIT Press (2008)
9. Bondarenko, A., Dung, P., Kowalski, R., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93 (1997) 63–101
10. García, A., Simari, G.: Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming* 4(1-2) (2004) 95–138
11. Prakken, H.: An abstract framework for argumentation with structured arguments. *Argument and Computation* (2010) 1–31
12. Black, E., Hunter, A.: A relevance-theoretic framework for constructing and deconstructing enthymemes. *Journal of Logic and Computation* 22(1) (2012) 55–78
13. Apothéloz, D.: The function of negation in argumentation. *J. of Pragmatics* (1993) 23–38
14. Wang, L., Cardie, C.: Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In: Proc. ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. (2014)
15. Kraus, S., Lehmann, S., Magidor, D.: Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44 (1990) 167–207
16. Tarski, A.: *Logic, Semantics, Metamathematics*. J. H. Woodger (ed), Oxford U. Press (1956)