Logical Representation and Analysis for RC-Arguments

Leila Amgoud
IRIT – CNRS
Toulouse – FRANCE
amgoud@irit.fr

Philippe Besnard
IRIT – CNRS
Toulouse – FRANCE
besnard@irit.fr

Anthony Hunter
University College London
London – U.K
anthony.hunter@ucl.ac.uk

***Abstract*—An argument is seen as *reasons* in favour of a *claim*. It is made of three parts: a set of *premises* representing the reasons, a *conclusion* representing the supported claim, and a *connection* showing how the premises lead to the conclusion.**

Arguments are frequently exchanged by human agents in natural language (spoken or written) in discussion, debate, negotiation, persuasion, etc. They may be very different in that their three components may have various forms.

In this paper, we propose a language for representing such arguments. We show that it is general enough to capture the various forms of arguments encountered in natural language, and that it is possible to represent attack and support relations between arguments as formulas of the same language.

***Keywords*-Arguments; Representation language.**

I. Introduction

An argument gives *reason* to support a *claim* that is questionable, or open to doubt. It is made of three parts: a set of *premises* representing the reason, a *conclusion* representing the supported claim, and a *link* showing how the premises lead to the conclusion [13]. The link is hence the *logical* part of an argument. The notion of argument is very rich and complex. Indeed, the reason (respectively the conclusion) varies from simple statements to combinations of arguments, and the link may be deductive, abductive, inductive, . . . [3]. Let us consider the following example of a natural language argument.

Example 1. *The title and first two paragraphs from an article on whether the London Heathrow airport should be expanded with a third runway. The article comes from the BBC website*[1].

> $\langle claim \rangle$ *Heathrow needs more capacity.* $\langle \backslash claim \rangle$
> $\langle reason \rangle$ *Heathrow runs at close to 100% capacity. With demand for air travel predicted to double in a generation, Heathrow will not be able to cope without a third runway, say those in favour of the plan.* $\langle \backslash reason \rangle$
> $\langle reason \rangle$ *Because the airport is over-stretched, any problems which arise cause knock-on delays. Heathrow, the argument goes, needs extra capacity if it is to reach the levels of service found at competitors elsewhere in Europe, or it will be overtaken by its rivals.* $\langle \backslash reason \rangle$

In the above tagging, we have a single claim, viz "Heathrow needs more capacity", and we have two reasons for this claim. Hence, we appear to have two arguments. Each with the same claim.

However, if we look at the second reason, we see that there are nested arguments, and so we could deconstruct the second paragraph as follows. The first sentence is an argument containing a reason and claim as follows

> $\langle reason \rangle$ *Because the airport is over-stretched* $\langle \backslash reason \rangle$, $\langle claim \rangle$ *any problems which arise cause knock-on delays.* $\langle \backslash claim \rangle$.

Then, we see that the above argument is itself a premise for the following claim

> $\langle claim \rangle$ *Heathrow, the argument goes, needs extra capacity if it is to reach the levels of service found at competitors elsewhere in Europe, or it will be overtaken by its rivals.* $\langle \backslash claim \rangle$

Putting these observations together, we could tag the second paragraph as follows where we have an argument as a nested reason.

> $\langle reason \rangle$ $\langle reason \rangle$ *Because the airport is over-stretched, any problems which arise cause knock-on delays.* $\langle \backslash reason \rangle$ $\langle claim \rangle$ *Heathrow, the argument goes, needs extra capacity if it is to reach the levels of service found at competitors*

[1] http://news.bbc.co.uk/1/hi/uk/7828694.stm

elsewhere in Europe, or it will be overtaken by its rivals. $\langle\backslash claim\rangle$ $\langle\backslash reason\rangle$

So the final paragraph contains an argument (i.e. a reason with claim), this argument is the reason for a claim within the paragraph. Furthermore, the whole paragraph is a reason for the claim in the title of the article.

Recently, there is growing interest in the computational models of argument community and in the computational linguistics community in mining arguments from texts (see for example [8], [9], [15], [16], [20], [21], [24], [25], [26], [27], [29], [31], [32]). An interesting challenge that is thus arising is the choice of target formalism for representing the extracted arguments. In computational models of argument, abstract argumentation (as proposed by Dung [14]) and logical or structured argumentation (as proposed in [5], [7], [17], [23]) are the two key options. Neither is ideal as a target formalism as we outline below.

Abstract argumentation: Each argument is atomic. There is no differentiation between reasons and claims. So there is insufficient structure for a target language for argument mining.

Logical argumentation: Each argument is a set of formulae for premises, and a formula for a claim, where the premises imply the claim using a given consequence operator of a particular monotonic logic. So there is excessive structure for a target language for argument mining since natural language arguments are generally enthymemes, that is some of their premises are unstated.

In previous work [1], we have proposed a formal language for representing arguments. The language is made of formulae of the form $(-)\mathcal{R}(y) : (-)\mathcal{C}(x)$ such that e.g., $\mathcal{R}(y) : \mathcal{C}(x)$ stands for y gives reason to claim x, $\mathcal{R}(y) : -\mathcal{C}(x)$ stands for y gives reason for not claiming x, and $-\mathcal{R}(y) : \mathcal{C}(x)$ stands for y is not a reason for claiming x. The latter form is called *rejection* of argument. The link between the reason and the claim is left implicit. We have shown that the language is general enough to capture a wide range of types of arguments. In this paper, we extend the language in such a way that the reason and/or the conclusion of an argument can also be a combination of arguments. This allows us to capture complex arguments like those conveyed in recommendation letters. Consider a recommendation letter written for someone who is applying for a position. The main claim is that the candidate deserves the position and the reason is a conjunction of several arguments, each of which may be nested. Another contribution of the paper consists of highlighting several forms of attacks and supports between arguments, some of them have never been defined in computational models of arguments. We show that each form can be defined as a formula of the language. This allows us to represent in a unified setting arguments and relations between them. We show also what issues in (computational) argumentation our formalism can address. In particular, we deal with the details of applying our formalism in a range of cases, discussing the significance of various forms of arguments allowed in our formalism.

The paper is organized as follows: Section II introduces the language. Section III shows how to encode attacks and supports between arguments as formulae of the language. Section IV compares our formalism with computational argumentation ones, and Section V describes briefly how the language can be used as a target language for argument mining. The last section concludes.

II. Syntax for RC formulae

Our formalism for representing arguments, inspired by Apothéloz [4], is built upon a classical propositional language $\mathbb{L}(\mathcal{A})$ where $\mathcal{A}$ is a set of atoms. The formulae of the language $\mathbb{L}(\mathcal{A})$ are defined in the usual way from $\mathcal{A}$ and the usual classical operators $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$. Our formalism also uses two functions $\mathcal{R}(.)$ and $\mathcal{C}(.)$, a disjunction operator $|$, a conjunction operator $\&$, and an additional negation operator $-$. Thus, two negation operators are needed: $\neg$ for denying propositional formulas ($\neg x$ denotes that x is false), and $-$ for denying $\mathcal{R}(.)$ and $\mathcal{C}(.)$. Please note that $\neg\neg x$ is identified with x and $--\mathcal{R}(.)$ is identified with $\mathcal{R}(.)$ (also, $--\mathcal{C}(.)$ is identified with $\mathcal{C}(.)$).

Definition 1 (RC formulae)**.** *The set of formulas* $\texttt{Arg}(\mathbb{L}(\mathcal{A}))$ *is the smallest set such that* a formula *is of the form* $(-)\mathcal{R}(y) : (-)\mathcal{C}(x)$ *where* x *and* y *are formulae of* $\mathbb{L}(\mathcal{A}) \cup \texttt{Arg}(\mathbb{L}(\mathcal{A}))$ *or is a Boolean combination of formulae of* $\texttt{Arg}(\mathbb{L}(\mathcal{A}))$ *with the connectives* $|$ *and* $\&$.

The two operators $|$ and $\&$ connect RC formulae as follows: $\mathcal{R}(y) : \mathcal{C}(x) \mid \mathcal{R}(z) : \mathcal{C}(t)$, $\mathcal{R}(y) : \mathcal{C}(x) \ \& \ \mathcal{R}(z) : \mathcal{C}(t)$, $\mathcal{R}(y) : \mathcal{C}(x) \mid -\mathcal{R}(z) : \mathcal{C}(t)$, $\mathcal{R}(y) : \mathcal{C}(x) \ \& \ -\mathcal{R}(z) : \mathcal{C}(t)$, ...

Remark: Please note that $-((-)\mathcal{R}(y) : (-)\mathcal{C}(x))$ is identified with $-(-)\mathcal{R}(y) : (-)\mathcal{C}(x)$.

Each formula is either an argument or a rejection of an argument. An *argument* is a reason for concluding a claim. It has two main parts: *premises* (the reason) and a *conclusion*. 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 (e.g., the premises cause the conclusion).

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

The functions $\mathcal{R}$ and $\mathcal{C}$ play the roles of *giving reason* and *concluding*, resp. They thus capture the coupling between a reason and a conclusion. As we will see later, the contents may be true while the functions do not hold and vice versa. The intuitive meaning of the two formal expressions captured by the previous definition is as follows:

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

The nature of the link between the reason and the conclusion is captured by the colon. There are at least two reasons for leaving the link implicit. First, natural language arguments are generally enthymemes, thus some of their premises are unstated. For instance, in the argument "Paul has DNA because he is human", there is a missing premise which says "All humans have DNA". Actually, it is not always possible to make the link explicit. The second reason is that there are several kinds of links, each of which leads to a particular definition of arguments. In [3], it was shown that arguments of types "threats" and "rewards" are defined in an abductive way, while arguments of type "appeals to prevailing practices" are deductive. Our purpose is to have one general definition of argument in which all the different types can be captured.

So far, the negation operator "$-$" has been used to deny the concluding function. In what follows, the function of *giving reason* can be denied as well by placing "$-$" 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 since it has exactly the opposite meaning of an argument.

Definition 3 (Rejection or anti-argument)**.** *A* rejection of an argument *is a formula of* $\mathtt{Arg}(\mathbb{L}(\mathcal{A}))$ *of the form* $-\mathcal{R}(y) : (-)\mathcal{C}(x)$.

The intuitive meaning for these formal expressions is as follows:

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

Example 2. *Assume the propositional atoms bird, penguin, damaged.wing (to denote animals with a damaged wing), slightly.damaged.wing (to denote animals with a slightly damaged wing), and egg.laying (to denote animals that lay eggs).*

1) $\mathcal{R}(bird) : \mathcal{C}(fly)$
2) $\mathcal{R}(penguin) : \mathcal{C}(\neg fly)$
3) $\mathcal{R}(bird \wedge damaged.wing) : -\mathcal{C}(fly)$
4) $-\mathcal{R}(bird \wedge slightly.damaged.wing) : -\mathcal{C}(fly)$
5) $-\mathcal{R}(egg.laying) : \mathcal{C}(fly)$
6) $-\mathcal{R}(egg.laying) : \mathcal{C}(\neg fly)$
7) $-\mathcal{R}(egg.laying) : -\mathcal{C}(fly)$

Arguments can be counterarguments for other arguments. For instance, $\mathcal{R}(bird) : \mathcal{C}(fly)$ *has* $\mathcal{R}(penguin) : \mathcal{C}(\neg fly)$ *and* $\mathcal{R}(bird \wedge damaged.wing) : -\mathcal{C}(fly)$ *as counterarguments. We investigate the notion of a counterargument as RC-formulae in the next section.*

III. Representing attack and support

In structured argumentation, an attack against a given argument consists of presenting another argument denying one of the components of the initial argument (i.e., premises, conclusion, link). Thus, similar to a rejection, the aim of an attack is to undermine an argument. The main difference between the two lies in that the attacker provides a reason for the attack. For instance, to undermine the conclusion x of an argument $\mathcal{R}(y) : \mathcal{C}(x)$, one should provide another argument justifying why $\neg x$ holds. In contrast, a rejection needs no justification (but it may have one). Every attack between two arguments leads to a rejection of the attacked argument in the following way:

If $\mathcal{R}(z) : \mathcal{C}(w)$ attacks $\mathcal{R}(y) : \mathcal{C}(x)$,
then $\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(w)) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))$.

Note that the converse is not true, rejections might not be transformed into an attack between a pair of arguments. Consider the following dialogue.

Paul: Why are you late? (la).
Carla: Because I am late $\mathcal{R}(la) : \mathcal{C}(la)$
Paul: This is not a reason $-\mathcal{R}(la) : \mathcal{C}(la)$

In the example, Paul rejects Carla's argument without justifying why. In fact, he denies the fact that the argument can be circular.

We now turn to showing how to detect attacks between mined arguments, and how the language can be used to capture them within formulae of the language. Recall that an argument $\mathcal{R}(y) : \mathcal{C}(x)$ may be attacked on one of its components: conclusion, premises, the link.

There are two ways for undermining the conclusion of an argument: a strong way by showing that the negation of the conclusion holds, and a weak way by showing that the conclusion fails.

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \qquad \mathcal{R}(z) : \mathcal{C}(\neg x)}{\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(\neg x)) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))} \left(\textbf{Strong Rebuttal}\right)$$

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \qquad \mathcal{R}(z) : -\mathcal{C}(x)}{\mathcal{R}(\mathcal{R}(z) : -\mathcal{C}(x)) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))} \left(\textbf{Weak Rebuttal}\right)$$

Strong Rebuttal corresponds to the well-known *rebuttal* in existing argumentation formalisms.

Example 3. *Illustration for strong rebuttal: Nixon is a quaker* (nq) *and Nixon is a republican* (nr)*. Is Nixon a pacifist* (np)*?*

$$\frac{\mathcal{R}(nq) : \mathcal{C}(np) \qquad \mathcal{R}(nr) : \mathcal{C}(\neg np)}{\mathcal{R}(\mathcal{R}(nr) : \mathcal{C}(\neg np)) : \mathcal{C}(-\mathcal{R}(nq) : \mathcal{C}(np))}$$

Weak rebuttal captures somehow the so-called undercutting relation [22]. The basic idea is to block the application of a defeasible rule in some cases. Let us consider the following example:

Example 4. *The object is red* (re) *because it looks red* (lr)*. This argument is written in our formalism as* $\mathcal{R}(lr) : \mathcal{C}(re)$*. In existing argumentation systems like ASPIC and ASPIC+, the hidden assumption "Objects that look red are indeed red" is encoded as a defeasible rule. If the object is illuminated by red light* (il)*, then undercutting amounts to blocking the application of the rule, thus blocking its conclusion* re*. In our language, this is simply written as* $\mathcal{R}(il) : -\mathcal{C}(re)$*. Unlike Example 3, the reason in the counter-argument* (il) *needs not command that the negation of the conclusion in the attacked argument* $(\neg re)$ *holds.*

The premises of an argument may also be undermined in a strong or a weak way as follows:

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \qquad \mathcal{R}(z) : \mathcal{C}(\neg y)}{\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(\neg y)) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))} \left(\textbf{Strong Premise Attack}\right)$$

$$\frac{\mathcal{R}(y) : \mathcal{C}(x) \qquad \mathcal{R}(z) : -\mathcal{C}(y)}{\mathcal{R}(\mathcal{R}(z) : -\mathcal{C}(y)) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))} \left(\textbf{Weak Premise Attack}\right)$$

Strong Premise Attack amounts to the well-known Assumption-Attack in argumentation literature.

Example 5. *Illustration of strong premise attack: The weather is good* (gw) *so the bbq will be a success* (bs) *but weather reports predict rain* (ra)*.*

$$\frac{\mathcal{R}(gw) : \mathcal{C}(bs) \qquad \mathcal{R}(ra) : \mathcal{C}(\neg gw)}{\mathcal{R}(\mathcal{R}(ra) : \mathcal{C}(\neg gw)) : \mathcal{C}(-\mathcal{R}(gw) : \mathcal{C}(bs))}$$

The last component of an argument is the link between the premises and the claim. As already said, the link concerns the logical part of the argument, that is the inference pattern that is used in order to infer the conclusion from the premises. It can be denied in three ways as follows:

$\mathcal{R}(z) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(x))$ **(Strong Reason Attack)**

$\mathcal{R}(z) : -\mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x))$ **(Weak Reason Attack)**

$-\mathcal{R}(y) : \mathcal{C}(x)$ **(Pure Reason Attack)**

Example 6. *Consider the following abductive argument. If all dogs are mammals* (dm)*, then all dogs are animals* (da)*. All dogs are animals. Therefore, all dogs are mammals. This argument can be written as:* $\mathcal{R}((dm \to da) \wedge da) : \mathcal{C}(dm)$*. Note that in spite of the premises and the conclusion all being true, the argument is not valid. Indeed, it uses the inference pattern*

$$\frac{B \qquad \text{if } A \text{ then } B}{A}.$$

Of course, it may happen that A *is false although* B *and if* A *then* B *are true. In the same spirit, one may reject the initial argument by means of a mere rejection* $-\mathcal{R}((dm \to da) \wedge da) : \mathcal{C}(dm)$*. There are many circumstances where a rejection can be justified, though. It would here be of the form* $\mathcal{R}(z) : \mathcal{C}(-\mathcal{R}((dm \to da) \wedge da) : \mathcal{C}(dm))$ *with* z *a case making the previous pattern invalid (e.g. read* da *as "David is annoyed" and* dm *as "David moans", let* z *stand for a situation where David is only slightly annoyed so that* da *is not enough of a reason for* dm*).*

Example 7. *Consider the following argument.* 90% *of humans are right-handed* (hrh)*, therefore Paul is also right-handed* (prh)*. The argument, written as* $\mathcal{R}(hrh) : \mathcal{C}(prh)$*, might be rejected* $(-\mathcal{R}(hrh) : \mathcal{C}(prh))$ *because the link between the premise* hrh *and the conclusion* prh *is invalid.*

To sum up, with our logic of arguments, we can formalize and manipulate attacks explicitly within the logic (which is not possible in other formal systems of argumentation), and we have a wider range of attacks than are considered in other formal proposals for argumentation.

Unlike attacks which express negative links between arguments, *supports* express positive links. In existing literature (e.g., [11]), such links are captured by a binary relation defined on the set of arguments. In our approach, such an external relation is not needed since supports can be expressed as formulae of the language. Let us now look at various forms of support. An argument may support another argument by approving one of its components: premises, claim and link.

$$\frac{\mathcal{R}(z) : \mathcal{C}(y) \qquad \mathcal{R}(y) : \mathcal{C}(x)}{\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(y)) : \mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x))} \quad \left(\begin{array}{c}\textbf{Premise}\\ \textbf{Support}\end{array}\right)$$

$$\frac{\mathcal{R}(z) : \mathcal{C}(x) \qquad \mathcal{R}(y) : \mathcal{C}(x)}{\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(x)) : \mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x))} \quad \left(\begin{array}{c}\textbf{Claim}\\ \textbf{Support}\end{array}\right)$$

$$\mathcal{R}(\mathcal{R}(z) : \mathcal{C}(t)) : \mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x)) \quad \textbf{(Reason Support)}$$

$$\mathcal{R}(z) : \mathcal{C}(\mathcal{R}(y) : \mathcal{C}(x)) \qquad \textbf{(Reason Support)}$$

The first two relations have been presented in [10] but both forms of link support are new.

Example 8. *Consider the following dialogue.*

Paul: *Carl will pass his exams (pe). He is smart (sm).* $\mathcal{R}(sm) : \mathcal{C}(pe)$

John: *He is moreover well prepared (wp).* $\mathcal{R}(wp) : \mathcal{C}(pe)$

John's argument can also be interpreted in a sense to be captured by the following argument: $\mathcal{R}(\mathcal{R}(wp) : \mathcal{C}(pe)) : \mathcal{C}(\mathcal{R}(sm) : \mathcal{C}(pe))$

IV. Comparison

We have explained the language for our approach, and we now turn to comparing it with existing approaches in the literature.

A. Implicit representation of links

In almost all works on structured argumentation (e.g. [2], [5], [17], [23], [28]), an argument is a set of premises that, using a notion of *derivation*, will lead to a conclusion. As argued in the introduction, this definition needs a logical representation of all the premises. This is not viable for arguments mined from texts since they are enthymemes (and therefore lack the explicit representation of premises and/or claim). Moreover, in general, existing definitions capture only one type of argument (viz. deductive arguments) while in text or dialogue, analogical arguments are very common. Our approach captures enthymemes and a wide variety of types of argument (such as abductive and inductive).

B. Capturing links not relying upon inference

As mentioned above, in almost all works on structured argumentation, an argument relies on a notion of derivation linking the premises of the argument to its conclusion. However, there arguably exist arguments that do not involve an inference from premises to claim. An example is:

Smoking is a reason to get cancer.

Please observe that this is definitely an argument. Though, there is no default rule to infer "getting cancer" from "smoking", not even a weighted default-like rule: It does not seem right to express that smoking entails getting cancer with likelihood x, whatever x in $[0, 1]$.

In other words, such arguments fall outside the realm of the existing approaches to structured argumentation. Nonetheless, such an argument can be naturally expressed in the RC-formalism as:

$$\mathcal{R}(s) : \mathcal{C}(gc)$$

where of course we use s to mean "smoking" and gc to mean "getting cancer".

C. Arguing explicitly about ignorance

In our approach, it is possible to argue explicitly about ignorance. For instance, the RC-formula $\mathcal{R}(s) : -\mathcal{C}(fe)$ can represent the argument below:

Since Carl is very smart (s), we cannot conclude that he will fail his exams (fe).

Some approaches to structured argumentation (e.g., [2], [23]) prevent the conclusion fe from being deduced in an indirect way, using Pollock's undercutting [22] for blocking the application of defeasibles rules as is illustrated next.

Example 9. *Consider the facts $\mathcal{F} = \{sm, \neg wh\}$, the set of strict rules $\mathcal{S} = \{sm \rightarrow n\}$, the set of defeasible rules $\mathcal{D} = \{\neg wh \rightarrow fe\}$, where fe denotes Carl will fail his exams, wh denotes Carl worked hard, sm denotes Carl is smart. Let n refer to the defeasible rule $\neg wh \rightarrow fe$ and let n denote its non-application. So the strict rule $sm \rightarrow n$ actually means that $\neg wh \rightarrow fe$ does not apply in the context of sm. The following four arguments can be built:*

- $a_1 : (< sm >, sm)$
- $a_2 : (< \neg wh >, \neg wh)$
- $a_3 : (< sm, sm \rightarrow n >, n)$
- $a_4 : (< \neg wh, \neg wh \rightarrow fe >, fe)$

Argument a_3 is a Pollock undercutting of a_4. Using Pollock undercutting as the attack relation, $\{a_1, a_2, a_3\}$ is the only stable extension. So, the set of conclusions drawn from the theory at hand *is $\{sm, \neg wh, n\}$. As expected, fe is not inferred. There is however no argument expressing that sm is the main reason for not having fe. While this approach is worthwhile in reasoning, it is not natural in dialogues where agents provide arguments for blocking conclusions. Our solution (i.e., using an argument of the form $\mathcal{R}(sm) : -\mathcal{C}(fe)$) makes the connection explicit.*

Instead of using a succinct formula such as $\mathcal{R}(y) : -\mathcal{C}(x)$ where x and y are propositions, structured argumentation identifies two arguments A_1 and A_2 where the reason (premises) of A_1 include x, and the claim of A_2 is y. Furthermore, for structured argumentation, the claim of A_1, and the premises of A_2, need to be determined in order to have the attack defined (which can be problematic when they are not explicitly represented in the text or dialogue). Consider the following argument:

> *Since Carl is at the university (u), he cannot conclude whether his printer is delivered at home (de)*

that can be represented simply by $\mathcal{R}(u) : -\mathcal{C}(de)$ in our approach. Contrastedly, in structured argumentation, not even blocking the conclusion de is possible without further information being available. In structured argumentation, not only is there need for an argument A with some explicit premises and claim de, but, in addition, a counterargument B is needed that either attacks the premises of A' or the derivation of the claim de from those premises.

D. Complex arguments

Our approach supports the representation of nesting of reasons. This means an argument or a rejection of an argument can be used as a reason or as a claim in an argument or rejection of an argument. The premises and conclusion can also be a conjunction or disjunction of arguments. This provides a rich formalism for representing arguments and rejections of arguments as arising in texts and dialogues.

DefLog [28] offers a language for representing arguments that in some respects is similar to our Definition 2. It has reasons and claims. These are atomic, or for claims, they can be nested arguments. So unlike our approach, DefLog does not support Boolean formulae as reasons or claims, and DefLog does not support nested reasons. Also, DefLog does not support rejections of arguments.

In [6], [19], abstract argumentation has been extended with attacks on attacks, but this form of meta-argumentation does not support differentiation of reasons and claims, and it does not support rejection of arguments. Meta-argumentation has also been proposed in logic-based approaches to argumentation (such as [18], [30]). These allow arguments to appear in the premises or claims of other arguments. However, these approaches assume explicit representation of the logical formulae in the premises by which the claim is derived, and they do not support rejection of arguments. Also, they do not support arguing about ignorance.

E. Rejection of arguments

Our approach incorporates the representation of a rejection of an argument. This is different to a counterargument, as we have argued in Section III. Approaches to structured argumentation (such as [5], [7], [17], [23]) represent counterarguments, but there is no proposal that represents rejection of arguments.

F. Explicit representation of attacks

Our approach represents rejections of arguments, which are essential for representing diverse mined arguments. Consequently, an attack is represented as an explicit construct in the language. Attacks can also be justified (e.g. $\mathcal{R}(x) : \mathcal{C}(-\mathcal{R}(y) : \mathcal{C}(z))$. As said before, no other logic-based approach to modelling argumentation provides a language for expressing rejection of arguments and attacks in the object language.

A proposal for introducing support and attacks relations into the language is E-DeLP [12], but the formalism only allows reasons to support or to attack a claim, and the claim can only be a defeasible rule. It therefore seems unlikely to be a suitable target language for representing mined arguments.

V. TARGET LANGUAGE FOR ARGUMENT MINING

Next we consider how our formalism can be used as a target language for argument mining. Tagging is an important step in developing a natural language processing system (an NLP system). For this, we need to annotate a corpora of items of text that we use for training an NLP system (for instance based on statistical natural language processing and/or machine learning). The aim is that after training, the NLP system can automatically tag previously unseen items of text (i.e. items of text not used for training) correctly. Since there is often a subjective aspect to the tagging.

each of text is tagged by a number of people independently.

Example 10. *We return to Example 1. We use the tag x_1 to tag a string in the title, we use the tags y_1 and y_2 to tag strings in the first paragraph, and we use the tags z_1, z_2, and z_3 to tag strings in the second paragraph. For each tag p, $\langle p \rangle$ denotes the start of the string, and $\langle \backslash p \rangle$ denotes the end of the string. From the following tagged strings, we can obtained the RC-formulae $\mathcal{R}(y_1) : \mathcal{C}(x_1)$ and $\mathcal{R}(\mathcal{R}(\mathcal{R}(z_1) : \mathcal{C}(z_2)) : \mathcal{C}(z_3)) : \mathcal{C}(x_1)$.*

- *$\langle x_1 \rangle$Heathrow needs more capacity$\langle \backslash x_1 \rangle$*
- *$\langle y_1 \rangle$Heathrow runs at close to 100% capacity. With demand for air travel predicted to double in a generation, Heathrow will not be able to cope without a third runway$\langle \backslash y_1 \rangle$, say those in favour of the plan.*
- *$\langle z_1 \rangle$Because the airport is overstretched$\langle \backslash z_1 \rangle$, $\langle z_2 \rangle$any problems which arise cause knock-on delays$\langle \backslash z_2 \rangle$. $\langle z_3 \rangle$Heathrow, the argument goes, needs extra capacity if it is to reach the levels of service found at competitors elsewhere in Europe, or it will be overtaken by its rivals$\langle \backslash z_3 \rangle$.*

So, as a target language, RC-formulae capture the connection between reasons and claims, and as illustrated in Example 10, this connection can be nested. Furthermore, arguments and anti-arguments can be nested as reasons and claims.

VI. Conclusions

This paper deals with the definition of a formalism for representing natural language arguments. The formalism provides a wide range of benefits including: (1) Target language for arguments, mined from texts or dialogues, that is between abstract and logical argumentation; (2) Representation of any type of arguments in a unified setting (threats, rewards, examples, ...); (3) Representation of arguments in favour of ignorance; (4) Explicit representation of attacks and supports in the object language; (5) Practical representation of enthymemes; (6) Representation of rejections (anti-arguments); and (7) Nesting and combinations of arguments and rejections.

This paper builds on a previous work [1] which it extends in the following way. Whereas [1] focussed on the inference system for RC-formulae, this paper deals with a number of notions in argumentation that can be captured by our approach and which facilitate the representation and reasoning with arguments as arising in AI applications involving text and dialogue. Indeed, representation of attack and support, nature of the reason-claim link and its underlying requirements/conditions, ability to capture mere ignorance, difference between rejections and counter-arguments, are all topics in this paper that were not addressed in [1].

References

[1] L. Amgoud, Ph. Besnard, and A. Hunter. Representing and reasoning with arguments mined from texts and dialogues. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'2015)*, LNCS. Springer, 2015.

[2] L. Amgoud, M. Caminada, C. Cayrol, M. Lagasquie, and H. Prakken. Towards a consensual formal model: inference part. Technical report, Deliverable, ASPIC Project (IST-002307 funded by EU FP6), 2004.

[3] L. Amgoud and H. Prade. Handling threats, rewards, and explanatory arguments in a unified setting. *International Journal of Intelligent Systems*, 20(12):1195–1218, 2005.

[4] D. Apothéloz. The function of negation in argumentation. *J. of Pragmatics*, pages 23–38, 1993.

[5] Ph. Besnard and A. Hunter. *Elements of Argumentation.* MIT Press, 2008.

[6] G. Boella, D. Gabbay, L. van der Torre, and S. Villata. Meta-argumentation modelling I: methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[7] A. Bondarenko, P. Dung, R. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[8] E. Cabrio and S. Villata. Generating abstract arguments: a natural language approach. In *Proc. 4th C. Computational Models of Argument (COMMA'12)*, pages 454–461. IOS Press, 2012.

[9] E. Cabrio and S. Villata. A natural language bipolar argumentation approach to support users in online debate interactions. *Argument and Computation*, 4(3):209–230, 2014.

[10] C. Cayrol and M. Lagasquie. On the acceptability of arguments in bipolar argumentation frameworks. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'2005)*, pages 378–389, 2005.

[11] C. Cayrol and M. Lagasquie. Bipolarity in argumentation graphs: towards a better understanding. *Int. J. of Approx. Reas.*, 54:876–899, 2013.

[12] A. Cohen, A. García, and G. Simari. Backing and undercutting in defeasible logic programming. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, volume 6717 of *LNCS*, pages 50–61. Springer, 2011.

[13] S. Cottrell. *Critical Thinking Skills : Developing Effective Analysis and Argument*. Palgrave Macmillan, 2011.

[14] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[15] V. Feng and G. Hirst. Classifying arguments by scheme. In *Proc. 49th Annual Meeting of ACL: Human Language Technologies (HLT 11)*, pages 987–996, 2011.

[16] E. Florou, S. Konstantopoulos, A. Koukourikos, and P. Karampiperis. Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 49–54. ACL, 2013.

[17] A. García and G. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[18] A. Hunter. Reasoning about the appropriateness of proponents for arguments. In *Proceedings of AAAI'08*, pages 89–94. MIT Press, 2008.

[19] S. Modgil and T. Bench-Capon. Metalevel argumentation. *Journal Logic and Computation*, 21(6):959–1003, 2011.

[20] M. Moens, E. Boiy, R. Mochales Palau, and C. Reed. Automatic detection of arguments in legal texts. In *Proc. 11th I. C. Art. Int. & Law (ICAIL'07)*, pages 225–230. ACM Press, 2007.

[21] A. Peldszus and M. Stede. From argument diagrams to argumentation mining in texts: a survey. *Int. Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31, 2013.

[22] J. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57:1–42, 1992.

[23] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, pages 1–31, 2010.

[24] N. Rooney, H. Wang, and F. Browne. Applying kernel methods to argumentation mining. In *Proc. 25th Int. Florida Art. Int. Research Society Conf. (FLAIRS'12)*, pages 272–275, 2012.

[25] J. Schneider, T. Groza, and A. Passant. A review of argumentation for the social semantic web. *Semantic Web: Interoperability, Usability, Applicability*, 4(2):159–218, 2013.

[26] Ch. Stab and I. Gurevych. Annotating argument components and relations in persuasive essays. In *Proc. 25th Int. Conf. Computational Linguistics (COLING'14)*, pages 1501–1510, 2014.

[27] S. Teufel, J. Carletta, and M. Moens. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of EACL 1999*, 1999.

[28] B. Verheij. Deflog: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13(3):319–346, 2003.

[29] L. Wang and C. Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In *Proceedings of the ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2014.

[30] M. Wooldridge, P. McBurney, and S. Parsons. On the meta-logic of arguments. In *ArgMAS*, pages 42–56, 2005.

[31] A. Wyner, R. Mochales Palau, M. Moens, and D. Milward. Approaches to text mining arguments from legal cases. In *Semantic Processing of Legal Texts*, volume 6036 of *LNCS*, pages 60–79. Springer, 2010.

[32] A. Wyner, J. Schneider, K. Atkinson, and T. Bench-Capon. Semi-automated argumentative analysis of online product reviews. In *Proc. 4th Conf. on Computational Models of Argument (COMMA'12)*, pages 43–50. IOS Press, 2012.