Argument Revival in Annotated Argumentation Networks

Abstract. This work explores the revival of arguments in abstract argumentation theory. A revived argument is an argument that is put forward in a specific context, but is somehow put aside because it did not seem relevant to the discussion or it has been denied, and then becomes useful in another context when a similar argument emerges. We obtain necessary conditions for argument revival using real-world examples and show that a recently proposed temporal argumentation framework [1] cannot account for argument revival. Next, we propose an algorithm for argument revival that uses arguments annotated by a context label. This algorithm determines what arguments can be revived using the context neighborhood, the attack relations and a revival policy.

Keywords: abstract argumentation theory, temporal networks, argument revival

1 Introduction

Argumentation represents a powerful paradigm to formalize commonsense reasoning. An abstract argumentation framework (AF) consists of a set of abstract arguments (which can for instance represent data or simply a proposition), and the conflicts between arguments, which are represented by a binary relation on the set of arguments [2]. To decide if an argument can be accepted or not, or if several arguments can be accepted together, Dung [3] defined several semantics of acceptance that allow, given an argumentation system, to compute sets of arguments, called extensions.

Several argument-based formalisms have emerged from this basic framework and have been used in autonomous agents and multi-agent systems. An agent can use argumentation for different reasons, for instance, it can perform individual reasoning to resolve conflicting information or to decide between conflicting goals [4, 5]; Agents in a multi-agent system use dialectical argumentation to identify and agree on differences between themselves, through interactions such as dialogs, negotiation, and agreement technology [6].

Recent research has introduced Temporal Argumentation Frameworks (TAF) extending Dung's AF with the consideration of argument's temporal availability [7, 8]. In TAF, arguments are valid only during specific time intervals (called availability intervals). Even though arguments in TAF are only available on certain time intervals, their attacks are assumed to be static and permanent over these intervals. This framework is later extended to Extended Temporal Argumentation Framework (E-TAF), enriching a TAF with the capability of modeling the availability of attacks among arguments [9]. For convenience, we will from now on refer to this framework as the BLCS framework, referring to the initials of the authors of the paper.

Although such temporal argumentation frameworks allow an agent to reason about arguments and attack relations in time, they do not allow modeling any *dynamics* between arguments of different time points. Still, both from a philosophical and practical point of view it is interesting to consider such dynamics. For instance, intuitively it make sense to state that arguments that were put forward in a specific moment in time, but were somehow put aside because they did not seem relevant to the discussion at the time or they were denied, can become useful in a later moment in time when a similar argument emerges.

In this paper we will study this phenomenon, which we coin *argument revival*[1], in more detail using abstract argumentation. By analyzing multiple real-world examples involving arguments are being revived, we obtain necessary conditions that any procedure for argument revival should meet, which we call the *revival conditions*. These conditions express that reviving should be based on some context and relevance relation (i.e., some attack relation), that reviving is an iterative process, and that new arguments can be revived based on status of arguments that are currently active (i.e., the extension). We show that existing formalisms do not meet these conditions and propose a framework that does fulfill them.

This framework consists of an annotated argumentation network and a revival algorithm. In an annotated argumentation network, arguments are annotated by a context. We implement the revival conditions into the revival algorithm, and we show that it gives intuitive results for several examples.

The rest of this paper is organized as follows: We start by introducing basic notions of (timed) argumentation in Section 2. Next, we characterize argument revival using several examples in Section 3. In Section 4, we first show that the BLCS framework cannot account for revival, then propose our argument revival algorithm and give an example of an instantiation of the model.

2 Preliminaries

2.1 Abstract Argumentation

Dung [3] introduced the notion of *Argumentation Framework* (AF) as a convenient abstraction of a defeasible argumentation system. In the AF, an argument is considered as an abstract entity with unspecified internal structure, and its role in the framework is completely determined by the relation of attack it maintains with other arguments.

Definition 1 (Argumentation Network (AF) [3]). *An* argumentation network *has the form $A = \langle S, R\rangle$, where S is a non-empty set of arguments and $R \subseteq S \times S$ is the attack relation. We may also write $x \rightarrow y$ to denote "x attacks y", i.e. for $(x,y) \in R$.*

Given an argumentation network A, an argument $x \in S$ is considered *acceptable* if it can be defended by arguments in A against all arguments in A that attack it (attackers). This intuition is originally presented in [3]. In this work, though, we will use Caminada labeling as the semantics for the extensions of argumentation networks.

[1]Oxford dictionary: *revival* (noun): 1. (...), 2. an instance of something becoming popular, active, or important again: *cross-country skiing is enjoying a revival*

Definition 2 (Caminada Labeling [10]). *A* caminada labeling *for an argumentation network $A = \langle S, R\rangle$ is a function $\lambda : S \mapsto \{0, \frac{1}{2}, 1\}$, where ($0 =$ "out", $\frac{1}{2} =$ "undecided", $1 =$ "in") such that the following holds: Let $x, y_1, ..., y_n \in S$ be arguments such that $y_1, \ldots, y_n$ all attack x, i.e. $y_1 \to x, ..., y_n \to x$. Then,*

(a) $\lambda(x) = 1$ *if for all* $y_i, \lambda(y_i) = 0$
(b) $\lambda(x) = 0$ *if for some* $y_i, \lambda(y_i) = 1$
(c) $\lambda(x) = \frac{1}{2}$ *if for some* $y_i, \lambda(y_i) = \frac{1}{2}$ *and for all* $y_i, \lambda(y_i) \le \frac{1}{2}$.

It is possible to write conditions (a)-(c) in one equation [11]:

$$(\lambda^*) : \lambda(x) = 1 - \max_i\{\lambda(y_i)\}. \tag{1}$$

2.2 Temporal Argumentation

Recently, the work of Dung was extended by annotating arguments with time intervals, and calculating for a given time interval the frames and extensions for that interval [7, 8]. A *Timed Abstract Framework* (TAF) is an extension of Dung's formalism where arguments are active (available) only during specific intervals of time; these intervals are called availability intervals. Attacks between arguments are considered *only* when both the attacker and the attacked arguments are available. Thus, when identifying the set of acceptable arguments the outcome associated with a TAF may vary in time.

A *time interval*, a period of time without interruptions, is represented as an interval $[a - b]$, where a and b are positive real numbers. For convenience, it is considered that a time intervals is a set of reals (denoted $\mathbb{R}$). A time intervals set is a subset $S \subseteq \mathbb{R}$. For instance, $\{[1 - 3], [4.5 - 8]\}$ is a time interval set. Using this notion we can define a *timed argumentation framework*.

Definition 3 (Timed Argumentation Framework [7]). *A* Timed Argumentation Framework *(or TAF) is a 3-tuple $\langle S, R, Av\rangle$*[2] *where S is a set of arguments, R is a binary relation defined over S and Av is an availability function for timed arguments, defined as $Av : S \to \wp(\mathbb{R})$, such that $Av(x)$ is the set of availability intervals of an argument x.*

Example 1 (Timed Argumentation Network). Consider the *TAF* $\Phi = \langle S, R, Av\rangle$ where:
$S = \{A, B, C, D, E, F, G\}$
$R = \{(B, A), (C, B), (E, A), (G, E), (F, G), (G, D)\}$
$Av = \{(A, \{[10-50], [80-120]\}); (B, \{[55-100]\}); (C, \{[40-90]\}); (D, \{[10-30]\}); (E, \{[20-75]\}); (F, \{[5-30]\}); (G, \{[10-40]\})\}$ (See Figure 1[3]).

Acceptability in a TAF is defined as extensions of the acceptability notions for Dung argumentation frameworks using timed argument profiles or *t-profiles*.

[2]Note that in the original paper the name of the arguments set was *Ar* and the name of the attack relation predicate was *Attacks*. We have changed these to be consistent with the other literature but this does not change anything for the definitions.

[3]Conforming to the definition, the time intervals associated with the arguments are sets of intervals, but the set notation has been omitted for readability.

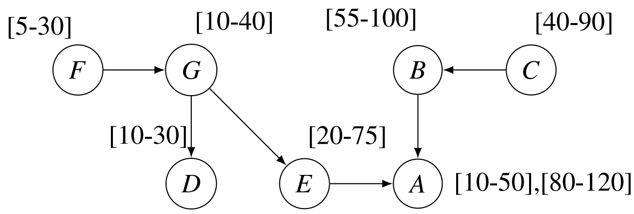

Fig. 1: Example network used in the BLCS framework paper [7].

Definition 4 (T-Profile [7]). *Let $\Phi = \langle S, R, Av \rangle$ be a TAF. A* timed argument profile *in Φ, or just* t-profile*, is a pair $\rho = (x, \tau)$ where $x \in S$ and τ is a set of time intervals; $(x, Av(x))$ is called the* basic *t-profile of argument x.*

Then, each argument can be assigned with a set of *acceptable* t-profiles with relation to a set of arguments. Since an argument must be defended against all its attackers that are considered acceptable, an argument is available only during time intervals in which it is defended against each of its attackers. This naturally leads to a partitioning of time intervals, depending on the attack relations between arguments. We can illustrate this by returning to our example.

Example 2 (Timed Argumentation Network, continued). Consider the network in Figure 2, which is the network that results after partitioning the time intervals for each argument in the network of Figure 1 and treating each separate t-profile as a separate argument.

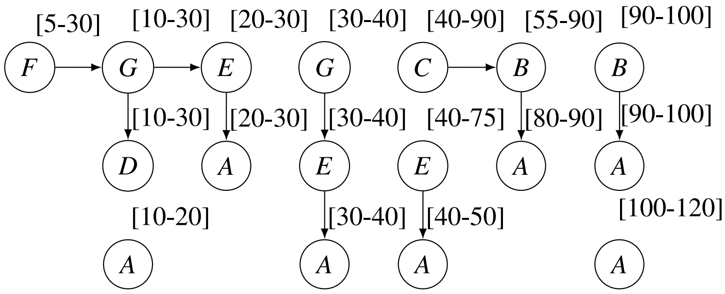

Fig. 2: Example network of the BLCS framework expanded for time intervals.

To calculate the acceptability of an argument in the *TAF*, we simply calculate the Dung extensions for this network and collect all annotations. For instance, suppose we want to calculate the acceptability of A. First notice that the unique Dung extension of this network is $\{A[10-20], A[30-40], A[80-90], A[100-120], B[90-100], C[40-90], D[10-30], E[20-30], E[40-75], F[5-30], G[30-40]\}$. The acceptability in A is then union of all time intervals in which A is valid in this extension, i.e. the set $\{[10-20], [30-40], [80-90], (100-120]\}$.

In [8], a TAF is extended by also considering the availability of attacks. An *Extended TAF* ($E-TAF$) is a 4-tuple $\langle S, R, Av, At\rangle$[4] were $\langle S, R, Av\rangle$ is TAF and $At : R \to \wp(\mathbb{R})$ with the following restriction $At((x,y)) \subseteq Av(x) \cap Av(y)$. Hence the availability of an attack can not exceed the availability of the arguments involved in the attack.

Also the definitions regarding *t-profiles*, *acceptability* and *Conflict freeness* are updated to the new network type, we omit them to keep the presentation simple.

3 Characterizing Argument Revival

In this section we discuss three examples of different real-world domains – historical analysis, tax investigation, and law – that contain arguments that are being revived. From these examples we infer four conditions, or characteristics, that we believe are minimally required for a formalism to model argument revival.

Example 3 (King George III). George III's (1738-1820) personal health was a concern throughout his long reign. He suffered from periodic episodes of physical and mental illness. In 1969, researchers asserted that these symptoms were characteristic of the disease porphyria, which was also identified in members of his family. In addition, a 2004 study of samples of the King's hair[5] revealed extremely high levels of arsenic, which is a possible trigger of disease symptoms. In 2005, it was suggested that the source of the arsenic could be the antimony used as a consistent element of the King's medical treatment [12].

In this example, there are three moments in time (in 1969, 2004, and 2005) in which new evidence puts previous arguments in a new daylight. For instance, the information that is put forward in 1969 – that George had the disease porphyria – is used in 2005 to motivate why the king had high levels of arsenic, which was discovered in 2004. So, arguments seem to be revived when are related in content (in terms of argumentation theory: there is an attack relation), but also when they have a temporal relation (i.e., one argument precedes another argument).

Example 4 (Tax investigation). Her Majesty's Revenue and Customs (HMRC) is a non-ministerial department of the UK Government responsible for the collection of taxes, the payment of some forms of state support and the administration of other regulatory

[4] Note that in the original paper the name of the availability function for timed arguments (Av) set was $ARAv$ and the name of the availability function for timed attacks (At) was $ATAv$. We have changed these to be consistent with our definitions.

[5] See "King George III: Mad or misunderstood?". BBC News. 2004-07-13. Retrieved 2010-04-25: http://news.bbc.co.uk/2/hi/health/3889903.stm

regimes including the national minimum wage[6]. A tax investigation for an individual or a business normal must be performed by the HMRC within 12 months of a tax return being submitted. If HMRC enquire into a tax return and find errors they can re-visit five years back. If they find errors in those years, they are allowed to re-visit five more years.

This example shows that reviving is an iterative process, where arguments are being revived *based on the status of the current arguments*; If the HMRC find errors (i.e., the argument that fraud has been detected is true), then the arguments that concern tax investigation in previous years can be revived, but if no fraud is detected it cannot.

Example 5 (McDonald v. Chicago lawsuit (2010)). Chicago resident Otis McDonald (76 year old), living in the Morgan Park neighborhood, wanted to purchase a handgun for personal home defense. Due to Chicago's requirement that all firearms in the city be registered, yet refusing all handgun registrations after 1982 when a citywide handgun ban was passed, he was unable to legally own a handgun. This case resulted in a landmark decision of the Court that the right of an individual to "keep and bear arms" protected by the Second Amendment applies to the states, overruling the local ban that Chicago enforced[7].

This last example shows that arguments can be revived not only based on time points, but on the scope of the arguments as well. The Chicago law dictates that hand gun possession is illegal, but during the law suit this local argument revives the global argument that keeping and bearing arms, as dictated in the Second Amendment of the United States, is allowed. From an argumentation theory point of view we can say that arguments are not only revived by time, but they can also be revived by something more general, which we call a *context*.

From this analysis we infer the necessary conditions that an argument revival framework should meet. These *revival conditions* are:

1. Arguments are annotated by a context (example 3), which is a generalization of time (example 1, 2);
2. Arguments can be revived based a context relation or an attack relation (example 1, 2, 3)
3. Revival is an iterative process: reviving an argument can create a chain of subsequently revived arguments (example 1,2).
4. Iterative revival is guided by some *revival policy*: Arguments can be revived based on the extension of the revived network (example 2);

4 Towards a Framework for Argument revival

In this section we will define a framework for argument revival, that consists of an annotated argumentation network and an algorithm for argument revival. First, we will show that the BLCS framework is not sufficient for reviving arguments.

[6] See http://www.hmrc.gov.uk/ for more information

[7] See http://en.wikipedia.org/wiki/McDonald_v._Chicago for a summary of this case, and http://www.chicagoguncase.com/case-filings/#SupremeCourt for the official case filings.

4.1 Argument revival in the BLCS framework

The BLCS framework considers arguments that are valid in a set of time points. This immediately invalidates the first revival condition of Section 3, stating that arguments should be annotated by a more general *context*. Still, this condition seems to be rather weak, and one could argue that argument revival over time intervals is simply a specific case of revival over contexts.

What is more problematic, though, is that the BLCS framework also invalidates the fourth revival condition. What became clear from the tax investigation is that it is quite conceivable that an argument will be revived depending on the validity of the current arguments, which means that in some case it can be revived, while in other cases it is not. This is not possible in the BLCS framework. Arguments within one time interval are considered all together, and there is no way to capture dynamics of revival using partial extensions.

Thus, in order to model argument revival we need a more general framework that uses argument contexts, and that can capture the dynamics of argument revival using some revival policy.

4.2 The Revival Framework

The framework for argument revival that we propose consists of an *annotated argumentation network* and an algorithm to compute revival. The annotated network contains arguments that are annotated by *contexts*, which are part of an algebra. Such an annotated network can be *instantiated* to a certain *context*, resulting in an ordinary argumentation network. We will start out by describing the context algebra.

Definition 5 (Context Algebra). *A* context algebra *is a set $\mathscr{C}$ containing elements called contexts, which includes the empty context 0 and the universal context 1. We define a commutative associative operation $\cap$ on contexts, that makes $\mathscr{C}$ into a lattice with operation $\cap$. This means that it has the following properties: $1 \cap X = X, 0 \cap X = 0, X \cap Y = Y \cap X$, and $X \cap (Y \cap Z) = (X \cap Y) \cap Z$.*

We define the preference between two contexts x and y as follows: $x \subseteq y =_{def} x \cap y = x$. Here $x \subseteq y$ denotes that x is weakly preferred to y, which we extend in the usual way: $x = y$ denotes that x and y are equally preferred, i.e. $x \subseteq y \wedge y \subseteq x$, and $x \subset y$ denotes that x is strictly preferred to y, i,e. $x \subseteq y \wedge x \neq y$.

We assume that for any subset of $C \subseteq \mathscr{C}$ of contexts the set $\bigcap C$ exists, and that $\bigcap C$ is the greatest lower bound (in $\subseteq$) of C.

Example 6 (Context Algebra). Let T be a flow of time. Let the algebra $\mathscr{C}$ be all sets t of all moments of time (i.e., $t \subseteq T$). Let $\cap$ be ordinary intersection. When we have an argument x and we annotate it with $t : x$, we mean that the argument x is valid (effective, available, true) at the moments of t. Thus, if argument x says “The economy is in recession”, this may be valid in the period 2008-2015, but hopefully not in 2016 and certainly not in 2007.

Now, the argument x may attack another argument y = “We must spend money on global warming”, but on the other hand, this argument y may not be available before, say, 2010, because such causes may not have been recognized yet!

Using a context algebra, we define an annotated network, which annotates arguments with a context.

Definition 6 (Annotated Network). *An* annotated network *has the form $N = \langle S, R, \mathscr{C}, d \rangle$, where $\langle S, R \rangle$ is an argumentation network, $\mathscr{C}$ is a context algebra, and $d : S \mapsto \mathscr{C}$ is a function associating with each argument $x \in S$ a context c. We denote this with $c : x$, or simply $d(x) : x$.*

We can obtain an ordinary argumentation network from an annotated network by isolating a context c and only considering those arguments that are valid in c. This means that for each argument x in the resulting ordinary network, the context $d(x)$ is fully contained in the context c that we want to isolate, i.e., $c \subseteq d(x)$[8]. We call this a *network instantiation* to the context c.

Definition 7 (Network Instantiation). *Let an annotated network $N = \langle S, R, \mathscr{C}, d \rangle$ be given. Let c be a context. Define the* network instantation *of N to context c as $N_c = \langle S_c, R_c \rangle$, such that $S_c = \{x \in S \mid c \subseteq d(x)\}$ and $R_c = R \restriction S_c$[9]. N_c is an ordinary network that is not annotated.*

Example 7 (Annotated Network). Consider the annotated network in Figure 3 containing 4 arguments. Argument a and c are both effective (valid) at time 10, b is valid at time 9, and d is valid at time 8. So at time 10, we have the network $N_{10} = \langle S_{10}, R_{10} \rangle$ with $S_{10} = \{a, c\}$ and $R_{10} = \varnothing$. At time 9 the network $N_9 = \langle S_9, R_9 \rangle$ consists of a single node, $S_9 = \{b\}$ and $R_9 = \varnothing$. At time $\{9, 10\}$, i.e. all nodes are valid at both 9 and 10, we have that there are no such arguments thus the network $N_{\{9,10\}} = \langle \varnothing, \varnothing \rangle$.

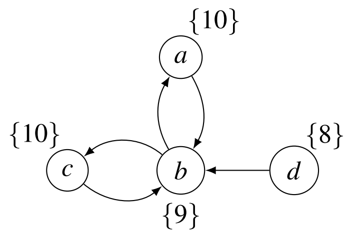

Fig. 3: Example annotated argumentation network.

[8]Note that we could also have chosen to turn this around and require that $d(x) \subseteq c$, meaning that each argument is active in at least a part of the context. We prefer to avoid this, because having arguments that are valid in only a subset of the context creates an additional layer of complexity because the network has to be decomposed further, similar to the decomposition that we showed in Figure 2.

[9]The operator $\restriction$ is a mathematical way of restricting the domain of the first argument to that of the second. Thus, $R \restriction S_c$ denotes that R is restricted to only those relations that have both elements in S_c.

Given an annotated node $c : x$, we define topological neighborhood $\mu(x)$ and the context neighborhood $\nu(c)$ for both the argument x and the context c, respectively.

Definition 8 (Topological neighbors). *Let $N = \langle S, R, \mathscr{C}, d\rangle$ be an annotated network. Let $d(x) : x$ be an annotated node with $x \in S$. The set $\mu(x)$ denotes the* topological neighbors *of x, i.e.: $\mu(x) =_{def} \{y \mid y = x \vee x \to y \vee y \to x\}$.*

Definition 9 (Context neighbors). *Let $N = \langle S, R, \mathscr{C}, d\rangle$ be an annotated network. Let $d(x) : x$ be an annotated node with $x \in S$. The set $\nu(x)$ denotes the* context neighbors *of x: $\nu(x) \subseteq \{y \mid y \subseteq x\}$, i.e. a subset of the arguments that have a more specific context than x.*

We do not fix a definition for the neighborhood function μ because this depends on the specific problem and the context algabra used. For example, in the case that we use time intervals for the annotation of the arguments as in the BLCS framework, we could define a revival period of 10% using the neighborhood function for these time intervals as:

$$\nu(d(x)) =_{def} \{y \mid d(y) = [\tau_1, \tau_2] \wedge d(x) \subseteq [\tau_2, \frac{\tau_2 - \tau_1}{10}]\}$$

Lastly, we introduce a *revival policy*, which allows one to decide what arguments are allowed to revive, based on the current extension of a network. We use the Caminada labeling to determine the extension (Definition 2).

Definition 10 (Revival Policy). *Let $N = \langle S, R, \mathscr{C}, d\rangle$ be an annotated network. Let $N' = \langle S', R'\rangle$ be a regular network such that $S' \subseteq S$ and $R' = R \upharpoonright S$, i.e., N' is an unannotated subset of N. Let $\lambda_{R'}$ be the Caminada labeling for network R' as defined in Definition 2. The* revival policy *for network N' is a mapping $\mathscr{P} : 2^S \times \lambda \to 2^S$ that takes a set of arguments and a labeling, and returns a subset of arguments, i.e. if $\mathscr{P}(S, \lambda) = S'$, then $S' \subseteq S$.*

4.3 Revival Algorithm

In the previous section we have provided a framework for argument revival, together with definitions that we will now use in the revival algorithm. The revival algorithm is an step-wise process, that revives arguments at each step based on the revival function, which we will define first.

Definition 11 (Revival Function). *An function $\mathcal{A} : \mathscr{C} \times S$ is said to be a* revival function *for an annotated network $N = \langle S, R, \mathscr{C}, d\rangle$ iff for every $c : x$ it returns a set of arguments as follows:*

$\mathcal{A}(c, x) \subseteq \{y \mid \nu(y) \in \nu(c) \vee y \in \mu(x)\}$.

Thus, for each $c \in \mathscr{C}$ and each $x \in S$ such that $c = d(x)$, the function $\mathcal{A}(c, x)$ returns a subset of the union of the topological neighbors of x, denoted by $\mu(x)$, and the context neighbors of c, denoted by $\nu(d(x))$. This set is called the revival set *associated with x,* *and we say that for any $y \in \mathcal{A}(c,x)$, the argument y was immediately revived because of x.*[10]

Definition 12 (Revival Algorithm). *Let $N = \langle S, R, \mathscr{C}, d, \mathcal{A}\rangle$ be a context annotated network with a revival function $\mathcal{A}$. Let c be a fixed context, and $N_c = \langle S_c, R_c\rangle$ be the network instantiation of N for context c as given in Definition 7. The* revival algorithm *generates a sequence Δ_c^m of revival arguments as follows:*

$$\Delta_c^0 = S_c$$
$$\Delta_c^{m+1} = \{y \mid x \in \mathscr{P}(S', \lambda_{N'}) \wedge y \in \mathcal{A}(d(x), x)\},$$
$$\text{where } N' = \langle S', R'\rangle \text{ s.t. } S' = \Delta_c^m \text{ and } R' = R \restriction \Delta_c^m$$

We define Δ_c as $\bigcup_m \Delta_c^m$.

Intuitively, a revival algorithm iteratively revives arguments in an annotated network starting from some context c. The first revival step Δ_c^0 simply contains all arguments that are valid for the context c, i.e. S_c. Then, at every revival step, arguments y are added to the new revival set Δ_c^{m+1} when they are revived by the revival algorithm $\mathcal{A}$. Note that arguments that are not selected by the revival policy $\mathscr{P}$ cannot revive other arguments. The set Δ_c contains all arguments that are active in the context c and all arguments that were revived using the revival algorithm. The resulting extension can then be determined from the network $\langle S^*, R^*\rangle$, where $S^* = \Delta_c$ and $R^* = R \restriction \Delta_c$.

It is clear that this algorithm meets all the conditions that we derived in Section 3. Arguments are annotated by a context (condition 1), and are iteratively revived (condition 3) based on the context or the attack relation (condition 2), or based on the current extension (condition 4).

4.4 Example

Due to space limitation, we will give a sketch of the implementation of the tax investigation example given in Section 3.

Let T be a flow of time. Let the algebra $\mathscr{C}$ be all sets t of all moments of time (i.e., $t \subseteq T$). We have an annotated argumentation network $N = \langle S, R, T, d, \mathcal{A}\rangle$ with arguments $S = \{err_1, \ldots, err_n, noerr_1, \ldots, noerr_k\}$, where each argument err_i represents the argument that an error has been found in the tax return in year i, while $noerr_j$ represents the argument that no error has been found in the year j. Thus, for each year i, we have $err_i \rightarrow noerr_i$ and $noerr_i \rightarrow err_i$ (see Figure 4).

We define the context neighborhood function μ (see Definition 9 as follows: $\mu(d(x)) = \{y \mid 0 \leq d(x) - d(y) \leq 5\}$, i.e., an argument can revive arguments back five years in time. Next, we define the revival policy to revive only if the argument err_i is *in*, i.e.

[10] Observe that this definition is very broad on purpose. We do not want to restrict the revival unless we have a good reason to. On the one side it should be possible to revive an argument based on the attack relations in the argumentation graph (that is, using μ), while on the other side it should be possible to revive an argument based on the context (that is, using ν).

$\lambda(err_i) = 1$, and the revival algorithm to revive arguments that are in the context neighborhood, which can only be revived by arguments that fulfill the revival policy.

Now, we can start from some context t. Initially, it is undecided whether any tax error have been made. If the tax investigator finds an error, the argument err_t will become *in*, which will revive all arguments $err_{t'}$ and $noerr_{t'}$ such that $0 \leq t - t' \leq 5$. Then, for each of these revived arguments it is undecided whether any errors have been found again, but once the tax investigator finds any error, all arguments going five years back are revived again.

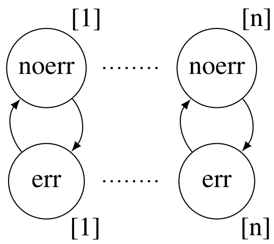

Fig. 4: Annotated network for the tax investigation example

5 Conclusion

In this paper we have introduced the notion of argument revival, which led to four conditions that characterized the real-life examples involving argument revival. Using these conditions, we proposed a framework for argument revival, and an algorithm that iteratively revives arguments within this framework. We implemented one of the real-life examples, which showed that our framework can be used to model such cases. In this conclusion, we would like to discuss how the concept of revival can be useful as a tool for knowledge representation in A.I.

Sharing knowledge is a requisite for communication, but in reality does not always lead to similar conclusions. This phenomenon can be explained with lack of rationality, partial information, uncertainty but also with interpretation. Interpretation is a fundamental process for knowledge representation; Differences between agents can be explained not only in the environment knowledge, but also in the way their mind work. Agent's cognitive models like BDI [13] or BODI [14] takes in acount those aspects but we are still lacking of interpretation mechanisms.

We can consider the problem of building an interpretation from shared representation as dialogical knowledge representation (an argumentation network). If each argument is annotated by some context (or a source), our revival algorithm can be used to build a network with arguments of different contexts using an agent's policy and a concept of neighborhood.

In future extensions we will stress parameters combinations exposing paradigmatical interpretations (parameters set-up patterns) in real-life cases. We will define a property of revival coherence: an evaluation criteria to evaluate a revived network according with agent's goals. Furthermore, we aim to explore the applicability of argument revival in knowledge representation, as we layed out in this conclusion.

References

1. M. Budán, M. Lucero, C. Chesñevar, and G. Simari, "An approach to argumentation considering attacks through time," in *Scalable Uncertainty Management* (E. Hllermeier, S. Link, T. Fober, and B. Seeger, eds.), vol. 7520 of *Lecture Notes in Computer Science*, pp. 99–112, Springer Berlin Heidelberg, 2012.
2. P. Besnard and A. Hunter, *Elements of Argumentation*. The MIT Press, 2008.
3. P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.
4. L. Amgoud and H. Prade, "Using arguments for making and explaining decisions," *Artif. Intell.*, vol. 173, pp. 413–436, Mar. 2009.
5. C. Beierle, B. Freund, G. Kern-Isberner, and M. Thimm, "Using defeasible logic programming for argumentation-based decision support in private law," in *Proceedings of the 2010 conference on Computational Models of Argument: Proceedings of COMMA 2010*, (Amsterdam, The Netherlands, The Netherlands), pp. 87–98, IOS Press, 2010.
6. P. Pasquier, R. Hollands, F. Dignum, I. Rahwan, and L. Sonenberg, "An empirical study of interest-based negotiation," in *Proceedings of the ninth international conference on Electronic commerce*, ICEC '07, (New York, NY, USA), pp. 339–348, ACM, 2007.
7. M. L. Cobo, D. C. Martínez, and G. R. Simari, "On admissibility in timed abstract argumentation frameworks," in *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, (Amsterdam, The Netherlands, The Netherlands), pp. 1007–1008, IOS Press, 2010.
8. M. L. Cobo, D. C. Martnez, and G. R. Simari, "Acceptability in timed frameworks with intermittent arguments.," in *EANN/AIAI (2)* (L. S. Iliadis, I. Maglogiannis, and H. Papadopoulos, eds.), vol. 364 of *IFIP Advances in Information and Communication Technology*, pp. 202–211, Springer, 2011.
9. M. C. Budán, M. J. G. Lucero, C. I. Chesñevar, and G. R. Simari, "Modelling time and reliability in structured argumentation frameworks," in *KR*, 2012.
10. M. Wu, Yining; Caminada, "A labelling-based justification status of arguments," *Studies in Logic*, vol. 3, pp. 12–29, 2010.
11. D. M. Gabbay, "Introducing equational semantics for argumentation networks," in *ECSQARU* (W. E. Liu, ed.), vol. 6717. 6717 of *Lecture Notes in Computer Science*, p. 1935, Springer, 2011.
12. T. M. Cox, N. Jack, S. Lofthouse, J. Watling, J. Haines, and M. J. Warren, "King george iii and porphyria: an elemental hypothesis and investigation.," *Lancet*, vol. 366, no. 9482, pp. 332–5, 2005.
13. A. S. Rao and M. P. Georgeff, "Modeling rational agents within a bdi-architecture," 1991.
14. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. W. N. van der Torre, "The boid architecture: conflicts between beliefs, obligations, intentions and desires," in *Agents*, pp. 9–16, 2001.
15. D. M. Barringer, H.; Gabbay, "Modal and temporal argumentation networks," in *Time for Verification - Essays in Memory of Amir Pnueli*, vol. 6200 of *Lecture Notes in Computer Science*, pp. 1–25, Springer LNCS, 2010.
16. D. M. Gabbay, *Meta-Logical Investigations in Argumentation networks*. Research Monograph, College publications, 2013.