Similarity Measures based on Compiled Arguments

Leila Amgoud[1] and Victor David[2]

[1] CNRS – IRIT, France
[2] Paul Sabatier University – IRIT, France
{leila.amgoud,victor.david}@irit.fr

Abstract. Argumentation is a prominent approach for reasoning with inconsistent information. It is based on the justification of formulas by arguments generated from propositional knowledge bases. It has recently been shown that similarity between arguments should be taken into account when evaluating arguments. Consequently, different similarity measures have been proposed in the literature. Although these measures satisfy desirable properties, they suffer from the side effects of being syntax-dependent. Indeed, they may miss redundant information, leading to undervalued similarity. This paper overcomes this shortcoming by compiling arguments, which amounts to transforming their formulas into clauses, and using the latter for extending existing measures and principles. We show that the new measures deal properly with the critical cases.

1 Introduction

Argumentation is a reasoning process based on the justification of claims by *arguments*, i.e., reasons for accepting claims. It has been extensively developed in Artificial Intelligence. Indeed, it was used for different purposes including decision making (eg. [1,2]), defeasible reasoning (eg. [3,4]), and negotiation [5,6].

Argumentation is also used as an alternative approach for handling inconsistency in knowledge bases [7,8,9]. Starting from a knowledge base encoded in propositional logic, arguments are built using the consequence operator of the logic. An argument is a pair made of a set of formulas (called support) and a single formula (called conclusion). The conclusion follows logically from the support. Examples of arguments are $A = \langle\{p \wedge q \wedge r\}, p \wedge q\rangle$, $B = \langle\{p \wedge q\}, p \wedge q\rangle$ and $C = \langle\{p, q\}, p \wedge q\rangle$. Once arguments are defined, attacks between them are identified and a semantics is used for evaluating the arguments, finally formulas supported by strong arguments are inferred from the base.

Some semantics, like h-Categorizer [7], satisfy the Counting (or Strict Monotony) principle defined in [10]. This principle states that each attacker of an argument contributes to weakening the argument. For instance, if the argument $D = \langle\{\neg p \vee \neg q\}, \neg p \vee \neg q\rangle$ is attacked by A, B, C, then each of the three arguments will decrease the strength of D. However, the three attackers are somehow similar, thus D will lose more than necessary. Consequently, the authors in [11] have motivated the need for investigating the notion of similarity between pairs of such logical arguments. They introduced a set of principles that a reasonable similarity measure should satisfy, and provided several measures that satisfy them. In [12,13,14] several extensions of h-Categorizer that take into account similarities between arguments have been proposed.

While the measures from [11] return reasonable results in most cases, it was shown in [15], that they may lead to inaccurate assessments if arguments are not *concise*. An arguments is concise if its support contains only information that is useful for inferring its conclusion. For instance, the argument A is not concise since its support $\{p \wedge q \wedge r\}$ contains r, which is useless for the conclusion $p \wedge q$. The similarity measures from [11] declare the two arguments A and B as not fully similar while they support the same conclusion on the same grounds ($p \wedge q$). Consequently, both A and B will have an impact on D using h-Categorizer. In [15], such arguments are cleaned up from any useless information by generating the concise versions of each argument, the measures from [11] are applied on concise arguments. However, these works fail to detect the full similarity between the two concise arguments B and C. In this paper, we solve the above issue by compiling arguments. The idea is to transform every formula in an argument's support into clauses. We extend the Jaccard-based similarity measures from [11] and show that new versions improve accuracy of similarity.

The article is organised as follows: Section 2 recalls the notions of logical argument and similarity measure. Section 3 introduces compilation of arguments. Section 4 extends some measures of similarity. Section 5 extends and propose new principles for similarity measures, and the last Section 6 concludes.

2 Background

2.1 Logical Concepts

Throughout the paper, we consider classical propositional logic $(\mathcal{L}, \vdash)$, where $\mathcal{L}$ is a propositional language built up from a *finite* set $\mathcal{P}$ of variables, the two Boolean constants $\top$ (true) and $\perp$ (false), and the usual connectives ($\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$), and $\vdash$ is the consequence relation of the logic. A literal is either a variable or the negation of a variable of $\mathcal{P}$, the set of all literals is denoted $\mathcal{P}^{\pm}$. Two formulas $\phi, \psi \in \mathcal{L}$ are *logically equivalent*, denoted by $\phi \equiv \psi$, iff $\phi \vdash \psi$ and $\psi \vdash \phi$.

A formula ϕ is in negation normal form (NNF) if and only if it does not contain implication or equivalence symbols, and every negation symbol occurs directly in front of an atom. Following [16], we slightly abuse words and denote by $\texttt{NNF}(\phi)$ the formula in NNF obtained from ϕ by "pushing down" every occurence of $\neg$ (using De Morgan's law) and eliminating double negations. For instance, $\texttt{NNF}(\neg((p \rightarrow q) \vee \neg t)) = p \wedge \neg q \wedge t$.

Let $\phi \in \mathcal{L}$, ϕ is in a conjunctive normal form (CNF) if is a conjunction of clauses $\bigwedge_i c_i$ where each clause c_i is a disjunction of literals $\bigvee_j l_j$. For instance $p \wedge (q \vee t)$ is in a CNF while $(p \wedge q) \vee t$ is not.

We denote by $\texttt{Lit}(\phi)$ the set of literals occurring in $\texttt{NNF}(\phi)$, hence $\texttt{Lit}(\neg((p \rightarrow q) \vee \neg t)) = \{p, \neg q, t\}$. The function $\texttt{Var}(\phi)$ returns all the variables occurring in the formula ϕ (e.g., $\texttt{Var}(p \wedge \neg q \wedge t) = \{p, q, t\}$).

A finite subset Φ of $\mathcal{L}$, denoted by $\Phi \subseteq_f \mathcal{L}$, is *consistent* iff $\Phi \nvdash \perp$, it is *inconsistent* otherwise. Let us now define when two finite sets Φ and Ψ of formulas are equivalent. A natural definition is when the two sets have the same logical consequences, i.e., $\{\phi \in \mathcal{L} \mid \Phi \vdash \phi\} = \{\psi \in \mathcal{L} \mid \Psi \vdash \psi\}$. Thus, the three sets $\{p, q\}$, $\{p \wedge p, \neg\neg q\}$, and $\{p \wedge q\}$

are pairwise equivalent. This definition is strong since it considers any inconsistent sets as equivalent. For instance, $\{p, \neg p\}$ and $\{q, \neg q\}$ are equivalent even if the *contents* (i.e. meaning of variables and formulas) of the two sets are unrelated (assume that p and q stand respectively for "bird" and "fly"). Furthermore, it considers the two sets $\{p, p \rightarrow q\}$ and $\{q, q \rightarrow p\}$ as equivalent while their contents are different as well. Clearly, the two rules "birds fly" and "everything that flies is a bird" express different information. Thus, the two sets $\{p, p \rightarrow q\}$ and $\{q, q \rightarrow p\}$ should be considered as different. Thus, in what follows we consider the following definition borrowed from [17]. It compares formulas contained in sets instead of logical consequences of the sets.

Definition 1 (Equivalent Sets of Formulas). *Two sets of formulas $\Phi, \Psi \subseteq_f \mathcal{L}$ are* equivalent*, denoted by $\Phi \cong \Psi$, iff there is a bijection $f : \Phi \rightarrow \Psi$ s.t. $\forall \phi \in \Phi$, $\phi \equiv f(\phi)$.*

Example 1. $\{p, p \rightarrow q\} \ncong \{q, q \rightarrow p\}$, $\{p, \neg p\} \ncong \{q, \neg q\}$, $\{p, q\} \ncong \{p \wedge q\}$ while $\{p, q\} \cong \{p \wedge p, \neg\neg q\}$.

Let us define the notion of *argument* as in [7].

Definition 2 (Argument). *An* argument *built under the logic $(\mathcal{L}, \vdash)$ is a pair $\langle \Phi, \phi \rangle$, where $\Phi \subseteq_f \mathcal{L}$ and $\phi \in \mathcal{L}$, such that:*

- *Φ is consistent,* *(Consistency)*
- *$\Phi \vdash \phi$,* *(Validity)*
- *$\nexists \Phi' \subset \Phi$ such that $\Phi' \vdash \phi$.* *(Minimality)*

An argument $\langle \Phi, \phi \rangle$ is trivial *iff $\Phi = \emptyset$ and $\phi \equiv \top$.*

Example 2. The three pairs $A = \langle \{p \wedge q \wedge r\}, p \wedge q \rangle$, $B = \langle \{p \wedge q\}, p \wedge q \rangle$ and $C = \langle \{p, q\}, p \wedge q \rangle$ are arguments.

Notations: $\mathtt{Arg}(\mathcal{L})$ denotes the set of all arguments that can be built in $(\mathcal{L}, \vdash)$. For any $A = \langle \Phi, \phi \rangle \in \mathtt{Arg}(\mathcal{L})$, the functions $\mathtt{Supp}$ and $\mathtt{Conc}$ return respectively the *support* ($\mathtt{Supp}(A) = \Phi$) and the *conclusion* ($\mathtt{Conc}(A) = \phi$) of A.

Note that the argument A in the above example is not concise since r is irrelevant for the argument's conclusion. In [15], the concise versions of arguments are computed using the following technique of *refinement*.

Definition 3 (Refinement). *Let $A, B \in \mathtt{Arg}(\mathcal{L})$ s.t. $A = \langle \{\phi_1, \ldots, \phi_n\}, \phi \rangle$, $B = \langle \{\phi'_1, \ldots, \phi'_n\}, \phi' \rangle$. B is a* refinement *of A iff:*

1. *$\phi = \phi'$,*
2. *There exists a permutation ρ of the set $\{1, \ldots, n\}$ such that $\forall k \in \{1, \ldots, n\}$, $\phi_k \vdash \phi'_{\rho(k)}$ and $\mathtt{Lit}(\phi'_{\rho(k)}) \subseteq \mathtt{Lit}(\phi_k)$.*

Let $\mathtt{Ref}$ be a function that returns the set of all refinements of a given argument.

Compared to the definition in [15], we extended the second constraint so that literals of $\phi'_{\rho(k)}$ are also literals of ϕ_k. The reason is that we would like the arguments $\langle\{p \wedge (p \vee q)\}, p \vee q\rangle$ and $\langle\{(p \vee \neg q) \wedge (p \vee q)\}, p \vee q\rangle$ can be refined into $\langle\{p \vee q\}, p \vee q\rangle$ while this was not possible in the original definition.

It is worth mentioning that an argument may have several refinements as shown in the following example.

Example 2 (Continued). The following sets are subset of refinement of these arguments.

- $\{\langle\{p \wedge q \wedge r\}, p \wedge q\rangle, \langle\{p \wedge q \wedge (p \vee q)\}, p \wedge q\rangle, \langle\{p \wedge q\}, p \wedge q\rangle\} \subseteq \mathtt{Ref}(A)$
- $\{\langle\{p \wedge q\}, p \wedge q\rangle, \langle\{p \wedge \neg\neg q\}, p \wedge q\rangle\} \subseteq \mathtt{Ref}(B)$
- $\{\langle\{p, q\}, p \wedge q\rangle, \langle\{(p \vee p) \wedge q\}, p \wedge q\rangle\} \subseteq \mathtt{Ref}(C)$

Every argument is a refinement of itself.

Proposition 1. *For any argument $A \in \mathtt{Arg}(\mathcal{L})$, $A \in \mathtt{Ref}(A)$.*

2.2 Similarity Measures

In [11], various measures have been proposed for assessing similarity between pairs of logical arguments. They extended existing measures which compare sets of objects including Jaccard measure [18]. Due to space limitation, in this paper we will focus only on the latter.

Definition 4 (Jaccard Measure). *Let X, Y be arbitrary sets of objects. if $X \neq \emptyset$ or $Y \neq \emptyset$, then*

$$s_{\mathtt{jac}}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}^{3}$$

If $X = Y = \emptyset$, then $s_{\mathtt{jac}}(X, Y) = 1$.

We recall below how the above measure is used in [11] for logical arguments.

Definition 5 (Jaccard Similarity Measure). *Let $0 < \sigma < 1$. We define $\mathtt{sim}^{\sigma}_{\mathtt{jac}}$ as a function assigning to any pair $(A, B) \in \mathtt{Arg}(\mathcal{L}) \times \mathtt{Arg}(\mathcal{L})$ a value $\mathtt{sim}^{\sigma}_{\mathtt{jac}}(A, B) =$*

$$\sigma.s_{\mathtt{jac}}(\mathtt{Supp}(A), \mathtt{Supp}(B)) + (1 - \sigma)s_{\mathtt{jac}}(\{\mathtt{Conc}(A)\}, \{\mathtt{Conc}(B)\}).$$

Example 2 (Continued). $\mathtt{sim}^{0.5}_{\mathtt{jac}}(A, B) = \mathtt{sim}^{0.5}_{\mathtt{jac}}(A, C) = \mathtt{sim}^{0.5}_{\mathtt{jac}}(B, C) = 0.5 \cdot 0 + 0.5 \cdot 1 = 0.5$.

[3] || stands for the cardinality of a set.

3 Compilation of arguments

Recall that $\{p, q\} \ncong \{p \wedge q\}$ while the two sets contain the same information. For getting them equivalent, we transform every formula into a CNF, then we split it into a set containing its clauses. Note that it is well known (eg. [19]) that any formula can be transformed into equivalent CNFs using the same literals. In our approach, we consider one CNF per formula. For that purpose, we will use a finite sub-language $\mathcal{F}$ that contains one formula per equivalent class and the formula should be in a CNF.

Definition 6 (Finite CNF language $\mathcal{F}$). *Let $\mathcal{F} \subset_f \mathcal{L}$ such that $\forall \phi \in \mathcal{L}$, there exists a unique $\psi \in \mathcal{F}$ such that:*

- $\phi \equiv \psi$,
- $\mathtt{Lit}(\phi) = \mathtt{Lit}(\psi)$,
- ψ *is in a CNF.*

Let $\mathtt{CNF}(\phi) = \psi$.

While we do not specify the elements of $\mathcal{F}$, we use concrete formulas in the examples, and they are assumed to belong to $\mathcal{F}$.

Notation: For $\Phi \subseteq_f \mathcal{L}$, $\mathtt{UC}(\Phi) = \bigcup\limits_{\phi \in \mathtt{CNF}(\Phi)} \; \bigcup\limits_{\delta \text{ clause } \in \phi} \delta$.

Let us now introduce the notion of compiled argument.

Definition 7 (Compiled Argument). *The* compilation *of $A \in \mathtt{Arg}(\mathcal{L})$ is* $A^* = \langle \mathtt{UC}(\mathtt{Supp}(A)), \mathtt{Conc}(A) \rangle$.

Example 2 (Continued). The compilations of the three arguments A, B, C are:

- $A^* = \langle \{p, q, r\}, p \wedge q \rangle$,
- $B^* = \langle \{p, q\}, p \wedge q \rangle$,
- $C^* = \langle \{p, q\}, p \wedge q \rangle$.

Note that the compilation of an argument is not necessarily an element of $\mathtt{Arg}(\mathcal{L})$ as it may violate the minimality condition. For instance, we can see that $A^* \notin \mathtt{Arg}(\mathcal{L})$ since the set $\{p, q, r\}$ is not minimal.

To solve this problem of non-minimal compiled arguments, we will extend the notion of concise argument (from [15]) while fixing the syntax-dependency issue. To define what a concise argument is, we first need to introduce what equivalent arguments are.

Thanks to the use of the language $\mathcal{F}$ in the compiled arguments, any equivalent formulas using the same literals have an identical syntax. Using this feature, we define that arguments are equivalent when they have identical compiled supports and conclusions.

Definition 8 (Equivalent Arguments). *Two arguments $A, B \in \mathtt{Arg}(\mathcal{L})$ are equivalent, denoted by $A \approx B$, iff*

$$\mathtt{UC}(\mathtt{Supp}(A)) = \mathtt{UC}(\mathtt{Supp}(B)) \textit{ and } \mathtt{UC}(\{\mathtt{Conc}(A)\}) = \mathtt{UC}(\{\mathtt{Conc}(B)\}).$$

We denote by $A \not\approx B$ when A and B are not equivalent.

Example 2 (Continued). $B \approx C$ while $A \not\approx B$ and $A \not\approx C$.

Note that arguments having the same compilation of their supports may not be equivalent. This is for instance the case of the two arguments D and E such that $D = \langle\{(p \vee q) \wedge (p \vee \neg q)\}, (p \vee q) \wedge (p \vee \neg q)\rangle$ and $E = \langle\{(p \vee q) \wedge (p \vee \neg q)\}, p\rangle$. Clearly, $D \not\approx E$.

We show that trivial arguments are not equivalent. For instance $\langle\emptyset, p \vee \neg p\rangle \not\approx \langle\emptyset, q \vee \neg q\rangle$ because $\{p \vee \neg p\} \neq \{q \vee \neg q\}$.

Proposition 2. *Trivial arguments are pairwise non equivalent.*

The objective of the notion of concise arguments is to produce the set of compiled arguments that satisfy the Definition 2. To do this, we observed that any non-equivalent refined argument A', of a compiled argument $A^* \in \texttt{Arg}(\mathcal{L})$, produces a new clause by inference between several clauses of the same formula.

Proposition 3. *Let* $A \in \texttt{Arg}(\mathcal{L})$*. If* $A' \in \texttt{Ref}(A)$*,* $A' \not\approx A$ *and* $A^* \in \texttt{Arg}(\mathcal{L})$ *then* $\exists \delta' \in \texttt{UC}(\texttt{Supp}(A'))$ *such that* $\forall \delta \in \texttt{UC}(\texttt{Supp}(A))$*,* $\delta \neq \delta'$*.*

Let us define a concise argument.

Definition 9 (Conciseness). *An argument* $A \in \texttt{Arg}(\mathcal{L})$ *is* concise *iff for all* $B \in \texttt{Ref}(A)$ *such that:*

- $B^* \in \texttt{Arg}(\mathcal{L})$ *and,*
- $\forall \delta \in \texttt{UC}(B)$*,* $\exists \delta' \in \texttt{UC}(A)$ *s.t.* $\delta = \delta'$*,*

we have $A \approx B$*.*
Let $\texttt{CR}(A)$ *denote the set of all* A' *concise refinements of* A *such that* $\texttt{UC}(\texttt{Supp}(A')) \subseteq \texttt{UC}(\texttt{Supp}(A))$*.*

The first constraint ensures that argument A does not have unnecessary information (clause) to infer its conclusion. The second constraint prevents argument A from being compared with arguments that have created new information (clause) in their support.

Example 2 (Continued). The following sets are subset of concise refinements of the arguments A, B, C:

- $\{\langle\{p \wedge q\}, p \wedge q\rangle, \langle\{p \wedge p \wedge q\}, p \wedge q\rangle\} \subset \texttt{CR}(A)$
- $\{\langle\{p \wedge q\}, p \wedge q\rangle, \langle\{p \wedge q \wedge q\}, p \wedge q\rangle\} \subset \texttt{CR}(B)$
- $\{\langle\{p, q\}, p \wedge q\rangle, \langle\{p \wedge p, q \wedge q\}, p \wedge q\rangle\} \subset \texttt{CR}(C)$

The non trivial argument have an infinite set of concise refinements.

Proposition 4. *Let* $A \in \texttt{Arg}(\mathcal{L})$*, if* A *is not trivial then* $|\texttt{CR}(A)| = \infty$*.*

A trivial argument have only one concise refinement, itself.

Proposition 5. *Let a trivial argument* $A \in \texttt{Arg}(\mathcal{L})$*,* $\texttt{CR}(A) = \{A\}$*.*

We show in the following theorem that the notion of a concise argument is equivalent to that of a compiled argument satisfying the conditions of the Definition 2.

Theorem 1. *For any $A \in \texttt{Arg}(\mathcal{L})$, A is concise iff $A^* \in \texttt{Arg}(\mathcal{L})$.*

When two arguments are equivalent they have equivalent concise refinements. In the same way that we define a set of equivalent formulae (Definition 1), we may define a set of equivalent arguments.

Definition 10 (Equivalent sets of Arguments). *Two sets of arguments $\Omega_A, \Omega_B \subseteq_f \texttt{Arg}(\mathcal{L})$, are equivalent, denoted by $\Omega_A \cong_{arg} \Omega_B$, iff there is a bijection $f : \Omega_A \to \Omega_B$ s.t. $\forall A' \in \Omega_A$, $A' \approx f(A')$. Otherwise Ω_A and Ω_B are not equivalent, denoted by $\Omega_A \ncong_{arg} \Omega_B$.*

Equivalent arguments have equivalent concise refinements.

Proposition 6. *Let $A, B \in \texttt{Arg}(\mathcal{L})$, if $A \approx B$ then $\texttt{CR}(A) \cong_{arg} \texttt{CR}(B)$.*

Example 2 (Continued). $B \approx C$ and $\texttt{CR}(B) \cong_{arg} \texttt{CR}(C)$.
We can also see in this example that there are non-equivalent arguments with equivalent concise refinements: $A \not\approx B$ and $\texttt{CR}(A) \cong_{arg} \texttt{CR}(B)$. This is because the equivalent arguments take into account irrelevant information.

Finally, note that when a compiled argument is not minimal, this may be due to the presence of unnecessary literals in the support which will be removed in the concise arguments; or because the argument is complex, i.e. it has different reasonings to conclude, which will produce different concise arguments.

Example 2 (Continued). Let $A = \langle \{p \wedge q \wedge r\}, p \wedge q \rangle, F = \langle \{p \wedge q, (p \to r) \wedge (q \to r)\}, r \rangle \in \texttt{Arg}(\mathcal{L})$.
The compiled argument of A and F are:

- $A^* = \langle \{p, q, r\}, p \wedge q \rangle$, and
- $F^* = \langle \{p, q, p \to r, q \to r\}, r \rangle$.

Clearly, r is irrelevant in A^* and F^* may infer r according to p or q. We may see in the concise versions of the arguments A and F that unnecessary information is removed and complex information is separated to have minimal arguments:

- $\{\langle \{p \wedge q\}, p \wedge q \rangle\} \subset \texttt{CR}(A)$, and
- $\{\langle \{p, p \to r\}, r \rangle, \langle \{q, q \to r\}, r \rangle\} \subset \texttt{CR}(F)$.

4 Extended Similarity Measures

We are ready now to introduce the new extended similarity measures.

First, we propose to apply similarity measure between set of objects on compiled supports and conclusions.

Definition 11 (Extended Jaccard Similarity Measure). *Let* $0 < \sigma < 1$. *We define* $\mathtt{sim}^{\sigma}_{\mathtt{jac}^*}$ *as a function assigning to any pair* $(A, B) \in \mathtt{Arg}(\mathcal{L}) \times \mathtt{Arg}(\mathcal{L})$ *a value* $\mathtt{sim}^{\sigma}_{\mathtt{jac}^*}(A, B) =$

$$\sigma.s_{\mathtt{jac}}(\mathtt{UC}(\mathtt{Supp}(A)), \mathtt{UC}(\mathtt{Supp}(B))) + (1-\sigma)s_{\mathtt{jac}}(\mathtt{UC}(\mathtt{Conc}(A)), \mathtt{UC}(\mathtt{Conc}(B))).$$

Example 2 (Continued).

- $\mathtt{sim}^{0.5}_{\mathtt{jac}^*}(A, B) = \mathtt{sim}^{0.5}_{\mathtt{jac}^*}(A, C) = 0.5 \cdot \frac{2}{3} + 0.5 \cdot 1 = \frac{5}{6} = 0.833.$
- $\mathtt{sim}^{0.5}_{\mathtt{jac}^*}(B, C) = 0.5 \cdot 1 + 0.5 \cdot 1 = 1.$

As we can see, working only with the compilation of arguments is not sufficient to assess degrees of similarity. It is also necessary to eliminate irrelevant information. In the following we propose to extend the two family from [15] dealing with concise refinements of an argument.

Definition 12 (Finite Conciseness). *Let* $A \in \mathtt{Arg}(\mathcal{L})$. *We define the set*

$$\overline{\mathtt{CR}}(A) = \{B \in \mathtt{CR}(A) \mid \mathtt{Supp}(B) \subset \mathcal{F}\}.$$

In this way, we obtain a finite set of non-equivalent concise refinements.

Proposition 7. *For every* $A \in \mathtt{Arg}(\mathcal{L})$, *the set* $\overline{\mathtt{CR}}(A)$ *is finite.*

We may now extend the two families of similarity measures (from [15]), to add a syntax independent treatment.

Definition 13 ($\mathtt{A}$-$\mathtt{CR}$ Jaccard Similarity Measure). *Let* $A, B \in \mathtt{Arg}(\mathcal{L})$, *and let* $\mathtt{sim}^{\sigma}_{\mathtt{jac}^*}$ *and* $\sigma \in]0, 1[$. *We define* $\mathtt{A}$-$\mathtt{CR}$ *Jaccard Similarity Measure*[4] *by* $\mathtt{sim}^{\mathtt{A}}_{\mathtt{CR}}(A, B, \mathtt{sim}^{\sigma}_{\mathtt{jac}^*}) =$

$$\frac{\sum\limits_{A_i \in \overline{\mathtt{CR}}(A)} \mathtt{Max}(A_i, \overline{\mathtt{CR}}(B), \mathtt{sim}^{\sigma}_{\mathtt{jac}^*}) + \sum\limits_{B_j \in \overline{\mathtt{CR}}(B)} \mathtt{Max}(B_j, \overline{\mathtt{CR}}(A), \mathtt{sim}^{\sigma}_{\mathtt{jac}^*})}{|\overline{\mathtt{CR}}(A)| + |\overline{\mathtt{CR}}(B)|}.$$

The value of $\mathtt{A}$-$\mathtt{CR}$ Jaccard Similarity Measure always belongs to the unit interval.

Proposition 8. *Let* $A, B \in \mathtt{Arg}(\mathcal{L})$, $\mathtt{sim}^{\sigma}_{\mathtt{jac}^*}$ *and* $\sigma \in]0, 1[$. *Then* $\mathtt{sim}^{\mathtt{A}}_{\mathtt{CR}}(A, B, \mathtt{sim}^{\sigma}_{\mathtt{jac}^*}) \in [0, 1]$.

Now we define our second family of similarity measures, which is based on comparison of sets obtained by merging supports of concise refinements of arguments. For an argument $A \in \mathtt{Arg}(\mathcal{L})$, we denote that set by

$$\mathtt{US}(A) = \bigcup_{A' \in \overline{\mathtt{CR}}(A)} \mathtt{UC}(\mathtt{Supp}(A')).$$

[4] The letter $\mathtt{A}$ in $\mathtt{A}$-$\mathtt{CR}$ stands for ”average”.

Definition 14 **($\texttt{U-CR}$ Jaccard Similarity Measure).** *Let $A, B \in \texttt{Arg}(\mathcal{L})$, $0 < \sigma < 1$, and $s_{\texttt{jac}}$. We define* $\texttt{U-CR}$ *Jaccard Similarity Measure*[5] *by* $\texttt{sim}^{\texttt{U}}_{\texttt{CR}}(A, B, s_{\texttt{jac}}, \sigma) =$

$$\sigma \cdot s_{\texttt{jac}}(\texttt{US}(A), \texttt{US}(B)) + (1 - \sigma) \cdot s_{\texttt{jac}}(\texttt{UC}(\texttt{Conc}(A)), \texttt{UC}(\texttt{Conc}(B))).$$

More generally, using the compilations allows to be more accurate even in the supports. Given that each clause are a formula the similarity degree can increase or decrease.

Example 3. Let $A = \langle\{p \wedge q, t\}, p \wedge q \wedge t\rangle$, $B = \langle\{p \wedge q, r\}, p \wedge q \wedge r\rangle$, $C = \langle\{p \wedge q, r\}, p \wedge q \wedge r\rangle$, $D = \langle\{s \wedge t \wedge u \wedge v, r\}, s \wedge t \wedge u \wedge v \wedge r\rangle \in \texttt{Arg}(\mathcal{L})$ and $\sigma = 0.5$.

- $\texttt{sim}^{0.5}_{\texttt{jac}}(A, B) = 0.5 \cdot \frac{1}{3} + 0.5 \cdot 0 = \frac{1}{6} = 0.167.$
- $\texttt{sim}^{0.5}_{\texttt{jac}^*}(A, B) = 0.5 \cdot \frac{2}{4} + 0.5 \cdot \frac{2}{4} = \frac{1}{2} = 0.5.$
- $\texttt{sim}^{0.5}_{\texttt{jac}}(C, D) = 0.5 \cdot \frac{1}{3} + 0.5 \cdot 0 = \frac{1}{6} = 0.167.$
- $\texttt{sim}^{0.5}_{\texttt{jac}^*}(C, D) = 0.5 \cdot \frac{1}{7} + 0.5 \cdot \frac{1}{7} = \frac{1}{7} = 0.143.$

Therefore $\texttt{sim}^{0.5}_{\texttt{jac}}(A, B) \leq \texttt{sim}^{0.5}_{\texttt{jac}^*}(A, B)$ and $\texttt{sim}^{0.5}_{\texttt{jac}}(C, D) \geq \texttt{sim}^{0.5}_{\texttt{jac}^*}(C, D)$.

Returning to our running example (from the introduction), we see that the three arguments A, B, C are now identified as being completely similar. That is, we have dealt with both the irrelevant information problem and the syntax-dependency problem.

Example 2 (Continued). For any $\sigma \in \ ]0, 1[$:

- $\texttt{sim}^{\texttt{A}}_{\texttt{CR}}(A, B, \texttt{sim}^{\sigma}_{\texttt{jac}^*}) = \texttt{sim}^{\texttt{A}}_{\texttt{CR}}(A, C, \texttt{sim}^{\sigma}_{\texttt{jac}^*}) = \texttt{sim}^{\texttt{A}}_{\texttt{CR}}(B, C, \texttt{sim}^{\sigma}_{\texttt{jac}^*}) = 1.$
- $\texttt{sim}^{\texttt{U}}_{\texttt{CR}}(A, B, s_{\texttt{jac}}, \sigma) = \texttt{sim}^{\texttt{U}}_{\texttt{CR}}(A, C, s_{\texttt{jac}}, \sigma) = \texttt{sim}^{\texttt{U}}_{\texttt{CR}}(B, C, s_{\texttt{jac}}, \sigma) = 1.$

The next Theorem ensure that the two family of similarity measures ($\texttt{A-CR}$ and $\texttt{U-CR}$) give the maximal degree of similarity between arguments only on arguments having equivalent concise refinements.

Theorem 2. *Let $A, B \in \texttt{Arg}(\mathcal{L})$, for any $\sigma \in \ ]0, 1[$, $\texttt{sim}^{\texttt{A}}_{\texttt{CR}}(A, B, \texttt{sim}^{\sigma}_{\texttt{jac}^*}) = \texttt{sim}^{\texttt{U}}_{\texttt{CR}}(A, B, s_{\texttt{jac}}, \sigma) = 1$ iff $\texttt{CR}(A) \cong_{arg} \texttt{CR}(B)$.*

From Proposition 6, we know that if $A \approx B$ then $\texttt{CR}(A) \cong_{arg} \texttt{CR}(B)$, then we can deduce the following corollary.

Corollary 1. *Let $A, B \in \texttt{Arg}(\mathcal{L})$, for any $\sigma \in \ ]0, 1[$, if $A \approx B$ then $\texttt{sim}^{\texttt{A}}_{\texttt{CR}}(A, B, \texttt{sim}^{\sigma}_{\texttt{jac}^*})$ $= \texttt{sim}^{\texttt{U}}_{\texttt{CR}}(A, B, s_{\texttt{jac}}, \sigma) = 1$.*

5 Extended Principles

The issue of syntax-dependence also exists in some principles. We propose here new principles for similarity measure between pairs of logical arguments and we extend some principles from [11].

The first new principle, called Minimality, ensures that similarity depends on the *content* of arguments. It states that if two arguments do not share any variables, then they are completely different. An example of such arguments are $\langle\{p\}, p \vee q\rangle$ and $\langle\{t\}, t\rangle$.

[5] $\texttt{U}$ in $\texttt{U-CR}$ stands for ”union”.

Principle 1 (Minimality) *A similarity measure* $\mathtt{sim}$ *satisfies* Minimality *iff for all* $A, B \in \mathtt{Arg}(\mathcal{L})$*, if*

– $\bigcup_{\phi_i \in \mathtt{Supp}(A)} \mathtt{Var}(\phi_i) \cap \bigcup_{\phi_j \in \mathtt{Supp}(B)} \mathtt{Var}(\phi_j) = \emptyset$ *and*
– $\mathtt{Var}(\mathtt{Conc}(A)) \cap \mathtt{Var}(\mathtt{Conc}(B)) = \emptyset$,

then $\mathtt{sim}(A, B) = 0$.

Example 4. Let $A = \langle \{p, q\}, p \wedge q \rangle, B = \langle \{r\}, r \rangle \in \mathtt{Arg}(\mathcal{L})$. A similarity measure $\mathtt{sim}$ satisfying Minimality ensure that $\mathtt{sim}(A, B) = 0$.

The second new principle consider that when two concise arguments have common information in their supports or conclusions hence they own some similarity between them.

Principle 2 (Non-Zero) *A similarity measure* $\mathtt{sim}$ *satisfies* Non-Zero *iff for all* $A, B, A^*, B^* \in \mathtt{Arg}(\mathcal{L})$*, if*

– $\mathtt{UC}(\mathtt{Supp}(A)) \cap \mathtt{UC}(\mathtt{Supp}(B)) \neq \emptyset$*, or*
– $\mathtt{UC}(\mathtt{Conc}(A)) \cap \mathtt{UC}(\mathtt{Conc}(B)) \neq \emptyset$,

then $\mathtt{sim}(A, B) > 0$.

Example 5. Let $A = \langle \{p, q\}, p \wedge q \rangle, B = \langle \{p\}, p \rangle \in \mathtt{Arg}(\mathcal{L})$. A similarity measure $\mathtt{sim}$ satisfying Non-Zero ensure that $\mathtt{sim}(A, B) > 0$.

The Maximality, Symmetry, Substitution, Syntax Independence principles defined in [11] do not need to be extend while (Strict) Monotony and (Strict) Dominance have to because they are dependent of the content.

Monotony states that similarity between two arguments is all the greater when the supports of the arguments share more formulas.

Principle 3 (Monotony – Strict Monotony) *A similarity measure* $\mathtt{sim}$ *satisfies* Monotony *iff for all* $A, B, C, A^*, B^*, C^* \in \mathtt{Arg}(\mathcal{L})$*, if*

1. $\mathtt{UC}(\mathtt{Conc}(A)) = \mathtt{UC}(\mathtt{Conc}(B))$ *or* $\mathtt{Var}(\mathtt{UC}(\mathtt{Conc}(A))) \cap \mathtt{Var}(\mathtt{UC}(\mathtt{Conc}(C))) = \emptyset$,
2. $\mathtt{UC}(\mathtt{Supp}(A)) \cap \mathtt{UC}(\mathtt{Supp}(C)) \subseteq \mathtt{UC}(\mathtt{Supp}(A)) \cap \mathtt{UC}(\mathtt{Supp}(B))$,
3. $\mathtt{UC}(\mathtt{Supp}(B)) \setminus \mathtt{UC}(\mathtt{Supp}(A)) \subseteq \mathtt{UC}(\mathtt{Supp}(C)) \setminus \mathtt{UC}(\mathtt{Supp}(A))$,

then the following hold:

– $\mathtt{sim}(A, B) \geq \mathtt{sim}(A, C)$ ***(Monotony)***
– *If the inclusion in condition 2 is strict or,* $\mathtt{UC}(\mathtt{Supp}(A)) \cap \mathtt{UC}(\mathtt{Supp}(C)) \neq \emptyset$ *and condition 3 is strict, then* $\mathtt{sim}(A, B) > \mathtt{sim}(A, C)$. ***(Strict Monotony)***

This extended monotony principle works only on concise arguments (i.e. using only relevant clauses). Indeed, when an argument has irrelevant information (e.g. $\langle \{p \wedge q \wedge r\}, p \wedge q \rangle$, r is irrelevant), this distorts the measurement.
In addition, constraints 2 and 3 are more precise than the original one. By making this compilation of supports we can go deeper into the formulas for a better evaluation.

Example 6. Let $A = \langle\{p \wedge q, r\}, p \wedge q \wedge r\rangle$, $B = \langle\{p, q, s\}, p \wedge q \wedge s\rangle$, $C = \langle\{p, s, (p \wedge s) \rightarrow t\}, t\rangle \in \mathtt{Arg}(\mathcal{L})$. Their compiled arguments are:

- $A^* = \langle\{p, q, r\}, p \wedge q \wedge r\rangle$,
- $B^* = \langle\{p, q, s\}, p \wedge q \wedge s\rangle$, and
- $C^* = \langle\{p, s, \neg p \vee \neg s \vee t\}, t\rangle$.

Thanks to the compilation of arguments, a similarity measure sim which satisfies the new principle of Strict Monotony, ensures that $\mathtt{sim}(A, B) > \mathtt{sim}(A, C)$.

Let us present the last principle dealing with the conclusions.

Principle 4 *[Dominance – Strict Dominance] A similarity measure* sim *satisfies* Dominance *iff for all* $A, B, C, A^*, B^*, C^* \in \mathtt{Arg}(\mathcal{L})$*, if*

1. $\mathtt{UC}(\mathtt{Supp}(B)) = \mathtt{UC}(\mathtt{Supp}(C))$,
2. $\mathtt{UC}(\mathtt{Conc}(A)) \cap \mathtt{UC}(\mathtt{Conc}(C)) \subseteq \mathtt{UC}(\mathtt{Conc}(A)) \cap \mathtt{UC}(\mathtt{Conc}(B))$,
3. $\mathtt{UC}(\mathtt{Conc}(B)) \setminus \mathtt{UC}(\mathtt{Conc}(A)) \subseteq \mathtt{UC}(\mathtt{Conc}(C)) \setminus \mathtt{UC}(\mathtt{Conc}(A))$,

then the following hold:

- $\mathtt{sim}(A, B) \geq \mathtt{sim}(A, C)$. ***(Dominance)***
- *If the inclusion in condition 2 is strict or,* $\mathtt{UC}(\mathtt{Conc}(A)) \cap \mathtt{UC}(\mathtt{Conc}(C)) \neq \emptyset$ *and condition 3 is strict, then* $\mathtt{sim}(A, B) > \mathtt{sim}(A, C)$. ***(Strict Dominance)***

Example 7. Let $A = \langle\{p \wedge q, r\}, p \wedge q \wedge r\rangle$, $B = \langle\{p, p \rightarrow q\}, p \wedge q\rangle$, $C = \langle\{p \wedge p \rightarrow q\}, q\rangle \in \mathtt{Arg}(\mathcal{L})$. Their compiled arguments are:

- $A^* = \langle\{p, q, r\}, p \wedge q \wedge r\rangle$,
- $B^* = \langle\{p, \neg p \vee q\}, p \wedge q\rangle$, and
- $C^* = \langle\{p, \neg p \vee q\}, q\rangle$.

Thanks to the compilation of arguments, a similarity measure sim which satisfies the new principle of Strict Dominance, ensures that $\mathtt{sim}(A, B) > \mathtt{sim}(A, C)$.

Theorem 3. *The three novel extended jaccard similarity measures satisfy all the principles.*

	$\mathtt{sim}^{\sigma}_{\mathtt{jac}}$	$\mathtt{sim}^{\sigma}_{\mathtt{jac}^*}$	$\mathtt{sim}^{\mathtt{A}}_{\mathtt{CR}}$	$\mathtt{sim}^{\mathtt{U}}_{\mathtt{CR}}$
Minimality	●	●	●	●
Non-Zero	○	●	●	●
Monotony	○	●	●	●
Strict Monotony	○	●	●	●
Dominance	○	●	●	●
Strict Dominance	○	●	●	●

The symbol ● (resp. ○) means the measure satisfies (resp. violates) the principle.
Table 1: Satisfaction of the principles of similarity measures

Note that the use of compilation in conclusions provides more accurate syntactic measures than in [11], as seen with the satisfaction of Strict Dominance.

Clearly, the original Jaccard measure is syntax-dependent and violates all the principles except Minimality (because without common literals, the syntax of the content does not matter).

Finally, we may also remark that the measure (Finally, we may also remark that the measure ($\mathtt{sim}^{\sigma}_{\mathtt{j}*}$) not taking into account concise arguments satisfies all the principles. This is due to the fact that the compiled arguments belong to the universe of possible arguments. We made this choice because a principle is a mandatory property. Thanks to this condition we keep the application of the principles general, by basing them on

$\mathtt{sim}^{\sigma}_{\mathtt{j}*}$) not taking into account concise arguments satisfies all the principles. This is due to the fact that the compiled arguments belong to the universe of possible arguments. We made this choice because a principle is a mandatory property. Thanks to this condition we keep the application of the principles general, by basing them on simple cases.

6 Conclusion

The paper further investigates the similarity between logical arguments. Based on the observation that existing similarity measures are syntax-dependent, they may provide inaccurate evaluations. We propose to compile the arguments in order to make the existing principles and similarity measures syntax-independent.

This work may be extended in several ways. The first is to identify a principle, or formal property, for distinguishing families of measures on concise arguments. The second is to use the new measures to refine argumentation systems that deal with inconsistent information. The third is to study the notion of similarity for other types of arguments, such as analogical arguments. The fourth is to study the usefulness of compiled arguments to produce simple and accurate explanations. Finally, we plan to study the notion of compiled arguments between other types of argument relations such as attacks and supports.

References

1. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence **173** (2009) 413–436
2. Zhong, Q., Fan, X., Luo, X., Toni, F.: An explainable multi-attribute decision model based on argumentation. Expert Systems Applications **117** (2019) 42–61
3. Governatori, G., Maher, M., Antoniou, G., Billington, D.: Argumentation semantics for defeasible logic. Journal of Logic and Computation **14**(5) (2004) 675–702
4. García, A., Simari, G.: Defeasible logic programming: an argumentative approach. Theory and Practice of Logic Programming **4**(1-2) (2004) 95–138
5. Sycara, K.: Persuasive argumentation in negotiation. Theory and Decision **28** (1990) 203–242
6. Hadidi, N., Dimopoulos, Y., Moraitis, P.: Argumentative alternating offers. In: Proceedings of the international conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10). (2010) 441–448

7. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence **128**(1-2) (2001) 203–235
8. Amgoud, L., Besnard, P.: Logical limits of abstract argumentation frameworks. Journal of Applied Non-Classical Logics **23**(3) (2013) 229–267
9. Vesic, S.: Identifying the class of maxi-consistent operators in argumentation. Journal of Artificial Intelligence Research **47** (2013) 71–93
10. Amgoud, L., Ben-Naim, J.: Axiomatic foundations of acceptability semantics. In: Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning KR. (2016) 2–11
11. Amgoud, L., David, V.: Measuring similarity between logical arguments. In: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning KR. (2018) 98–107
12. Amgoud, L., Bonzon, E., Delobelle, J., Doder, D., Konieczny, S., Maudet, N.: Gradual semantics accounting for similarity between arguments. In: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning KR. (2018) 88–97
13. Amgoud, L., David, V.: An adjustment function for dealing with similarities. In Prakken, H., Bistarelli, S., Santini, F., Taticchi, C., eds.: Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020. Volume 326 of Frontiers in Artificial Intelligence and Applications., IOS Press (2020) 79–90
14. Amgoud, L., David, V.: A General Setting for Gradual Semantics Dealing with Similarity. In: 35th AAAI Conference en Artificial Intelligence (AAAI 2021), Virtual Conference, United States, AAAI : Association for the Advancement of Artificial Intelligence, AAAI Press (February 2021)
15. Amgoud, L., David, V., Doder, D.: Similarity measures between arguments revisited. In Kern-Isberner, G., Ognjanovic, Z., eds.: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 15th European Conference, ECSQARU 2019, Belgrade, Serbia, September 18-20, 2019, Proceedings. Volume 11726 of Lecture Notes in Computer Science., Springer (2019) 3–13
16. Lang, J., Liberatore, P., Marquis, P.: Propositional independence-formula-variable independence and forgetting. Journal of Artificial Intelligence Research **18** (2003) 391–443
17. Amgoud, L., Besnard, P., Vesic, S.: Equivalence in logic-based argumentation. Journal of Applied Non-Classical Logics **24**(3) (2014) 181–208
18. Jaccard, P.: Nouvelles recherches sur la distributions florale. Bulletin de la Société Vaudoise des Sciences Naturelles **37** (1901) 223–270
19. Russell, S., Norvig, P.: Artificial intelligence: A modern appraoch (1995)