An Argumentation Approach for Resolving Privacy Disputes in Online Social Networks

NADIN KOKCIYAN, Bogazici University
NEFISE YAGLIKCI, Bogazici University
PINAR YOLUM, Bogazici University

Preserving users' privacy is important for Web systems. In systems, where transactions are managed by a single user, such as e-commerce systems, preserving privacy of the transactions is merely the capability of access control. However, in online social networks, where each transaction is managed by and has effect on others, preserving privacy is difficult. In many cases, the users' privacy constraints are distributed, expressed in a high-level manner and would depend on information that only becomes available over interactions with others. Hence, when a content is being shared by a user, others that might be affected by the content should discuss and agree on how the content will be shared online so that none of their privacy constraints are violated. To enable this, we model users of the social networks as agents that represent their users' privacy constraints as semantic rules. Agents argue with each other on propositions that enable their privacy rules by generating facts and assumptions from their ontology. Moreover, agents can seek help from others by requesting new information to enrich their ontology. Using Assumption-based Argumentation, agents decide whether a content should be shared or not. We evaluate the applicability of our approach on real-life privacy scenarios in comparison with user surveys.

CCS Concepts: •**Security and privacy** → **Social network security and privacy; Privacy protections; Usability in security and privacy;** •**Computing methodologies** → **Reasoning about belief and knowledge; Multi-agent systems; Intelligent agents;**

Additional Key Words and Phrases: Privacy, Agreement, Argumentation, Negotiation

ACM Reference Format:
Nadin Kökciyan, Nefise Yaglikci and Pınar Yolum, 2016. An Argumentation Approach for Resolving Privacy Disputes in Online Social Networks. *ACM Trans. Internet Technol.* V, N, Article 0000 (2016), 22 pages.
DOI: http://dx.doi.org/10.1145/3003434

1. INTRODUCTION

Online social networks allow their users to share content with others. Many times the shared content can be related to other users in the system. Generally, the person that puts up the content is said to own it and thus can decide who can see the content. For example, a user can take a picture of her party, tag some people and put it up on a social network. However, everybody has a private world and may want their pictures to be seen only by certain individuals. In such cases, currently they do not have many options but to either complain to the social network administration or ask the content owner to restrict the content to a certain audience. Yet, realistically this is too late as the content has already been published.

A preliminary version of this paper has been published as a short paper in AAMAS 2016. This work has been supported by TUBITAK under grant 113E543.
Author's addresses: N. Kökciyan, N. Yaglikci and P. Yolum, Department of Computer Engineering, Bogazici University, 34342 Bebek, Istanbul, Turkey.

DOI: http://dx.doi.org/10.1145/3003434

A recent study shows that if the content owners knew that some of their friends related to the content would be upset about it, the owner would have preferred to take steps to adjust the post to respect their privacy constraints [Stewart 2014]. That is, users would like to cooperate with friends to make sure they feel fine about the content being shared and resolve privacy disputes up front. Hence, this paper develops a method to enable users to discuss the privacy restrictions on a content before putting it up. There are two branches of work in the literature that are relevant. First, a set of approaches provide collaborative environments for the users to develop privacy policies together [Squicciarini et al. 2013]. While these could be considered as serving the same purpose, because it is done offline and by the users themselves, in practice they cannot be used frequently in an online social network. Second, a set of approaches advocate that agents, which represent the users, *negotiate* their privacy constraints in order to reach a consensus [Such and Rovatsos 2016; Mester et al. 2015]. Existing approaches in the second category mostly enable agents to exchange offers among themselves but do not enable agents to challenge each other's privacy constraints. Hence, while these approaches also enable a group decision to be reached, they do not allow agents to contemplate on their privacy constraints. However, ideally the interactions among agents need to be carried out in such a way that agents discuss their privacy constraints and try to resolve them so that each user's privacy is preserved as much as possible.

To enable this, we advocate an approach where each user in the social network is represented by an agent that keeps track of its user's privacy constraints. Each agent in the system is equipped with an ontology and the semantic rules that capture its user's privacy constraints. If a user decides to put up a content online, the user's agent first contacts all those relevant to the content (such as tagged or mentioned people in the content) to request permission. When the agents of these relevant individuals receive the request, they evaluate it using their rules to decide. If the content is acceptable, then the content can be shared online. However, if one of the agents has a concern (i.e., its privacy constraint is violated), then the requesting agent starts a persuasion dialogue as there is a conflict of opinions [Walton and Krabbe 1995]. Agents put forward *arguments* to defend their opinions and try to persuade other agents. Arguments are generated on demand from the agent's ontology as well as by consultation of other agents in the system. The requesting agent can continue the argumentation by providing further arguments. This continues in a turn-taking fashion and eventually when there are no more arguments to be provided, the existing arguments are evaluated in an argumentation system. While the general approach is applicable to any domain in which disputes can take place, we study it in the context of privacy.

Argumentation serves as the backbone of our approach. We use argumentation to simulate a persuasion dialogue between agents. Recently, Fan *et al.* succinctly showed that assumption-based argumentation (ABA) [Dung et al. 2009] can be used to conduct a dialogue for decision making [Fan et al. 2014]. We carry the result further. We use assumption-based argumentation in online social networks to enable users to engage in a dialogue to resolve privacy disputes over a post and to decide whether the post will be shared or not. Our main contributions are as follows:

- We develop a semantic agent representation that uses ontologies to represent domain knowledge and semantic rules to describe privacy constraints such that the semantic information can be used to perform argumentation.
- We provide an algorithm that enables agents to carry out a dialogue such that each agent can attack the assumptions of others by collecting necessary information (such as facts, assumptions, or rules) as necessary as well as using ontology inference to prepare attacks by proving contraries.

— We provide a working system that is implemented with Java and RESTful Web services that enables users to enter their privacy constraints, finds relevant agents to start the argumentation, and enables them to convince each other as to share or not share a particular post. Our system uses *abagraph* [Toni 2014] as an ABA engine to compute argumentation results.

The rest of this paper is organized as follows: Section 2 explains the technical framework for ABA and the agent architecture used with emphasis on the semantic rules for reasoning on privacy constraints. Section 3 shows how argumentation can be applied to resolve privacy conflicts. Section 4 describes our proposed approach PRIARG to show how agents can generate arguments in compliance with ABA and compute the result to decide on whether to share a given content. Section 5 evaluates the approach on real-life privacy scenarios to show its applicability as well as with comparison to similar approaches. Section 6 discusses the work in relation to the literature and provides directions for future work.

2. AGENT-BASED REPRESENTATION OF SOCIAL NETWORKS

It is becoming increasingly difficult to preserve privacy in online social networks. One major reason for this is that it is possible for anyone in the social network to share content that might be considered private for a second user. Since there is no single point of control, privacy needs to be preserved in a distributed manner. Consider the following five scenarios where Alice and Bob are in the same social network. Alice would like to share a picture where Bob is tagged. The picture shows a wristband that was given at Oktoberfest.

SCENARIO 1. Alice wants to share the post without consulting Bob.

SCENARIO 2. Alice wants to share the post but consults Bob first. Bob refuses because he is concerned that the wristband will signal his attendance to a festival.

SCENARIO 3. Following on Scenario 2, Alice has information that the wristband can also be found in a shop called $Gifty$; thus, the wristband would not necessarily imply Bob's attendance to a festival.

SCENARIO 4. Following on Scenario 3, Bob does not have any information to oppose to what Alice is saying. He consults another friend, who suggests that to be sure Bob should check if $Gifty$ is open. Bob thinks that the website of $Gifty$ cannot be accessed and comes to conclusion that $Gifty$ is out of business.

SCENARIO 5. Following on Scenario 4, Alice has the knowledge that $Gifty$ has another available website.

Scenario 1 depicts how most of the current online social networks work. The content owner shares the post independently, without consulting others in the picture. Scenario 2 resembles negotiation approaches in the literature in which the reason why an offer is refused is also provided [Mester et al. 2015; Amgoud et al. 2007]. Scenario 3, Scenario 4 and Scenario 5 reflect how a user can challenge an argument. For this, the user uses its own knowledge and consults others to create an argument.

In current online social networks, each scenario is executed as follows: Alice puts up the picture. Bob can either ask Alice to remove the picture and they can discuss offline about reasons or Bob can complain to the social network administration with reasons. However, it should be possible for these users to discuss in a structured manner *before* the picture is put up online and reach a conclusion if there is one.

2.1. Reasoning with Ontologies

We represent each user in the social network with a software agent. Agents are persistent computations that perceive, reason, act and communicate with other agents when necessary. The agent's main task is to manage its user's privacy constraints. We follow the previous work done using Semantic Web to represent users' privacy constraints and their social network [Carminati et al. 2011; Gandon and Sadeh 2004]. In line with the work of Mester *et al.* [Mester et al. 2015], each agent is equipped with an ontology to represent the social network as well as the privacy constraints of users.

2.1.1. A Social Network Ontology. A social network consists of *users* who are connected to other users via *relations* and share some *content* with a target audience. We use Web Ontology Language (OWL) to represent such a social network. In this ontology, concepts denote groups of instances (e.g. `Object` includes the instance `:wband`). We use object properties (e.g. *includesObject*) to relate instances and data properties (e.g. *isOrdinary*) to describe instance attributes[1].

An `Agent` sends a `PostRequest` to other agents. Each `PostRequest` is intended to be seen by a specific `Audience`, where *hasAudience* relates these two concepts. An audience is a group of agents, *hasMember* describes agents that are members of an audience. An agent may reject a post request, which is described via *rejects*. A `PostRequest` may contain some `Content` such as textual information `Text`, visual information `Medium` or `Location` information. *hasText*, *hasMedium* and *hasLocation* are used to relate corresponding concepts to `PostRequest`. A person may be mentioned in a text (*mentionedPerson*), tagged in a medium (*taggedPerson*) or at a specific location with other people (*withPerson*). In a social network, agents may be connected to other agents via various relationships. *isConnectedTo* is a property that connects an agent to another one. The sub-properties of *isConnectedTo* (*isColleagueOf*, *isFriendOf* and *isPartOfFamilyOf*) allow us to describe relations in more detail.

Many times privacy constraints rely heavily on the context of a post. However, the context of a post is difficult to judge even if the factual information such as time and location are available [Schmidt et al. 1999]. A picture taken in class may depict a student at a learning context and an instructor at a working context even though the time and location are the same. To capture the fact that users can have different privacy constraints based on context, we define various `Contexts` that can be associated with a post request. Each agent analyzes a post request and infers the context information according to its observations. Following the above example, a post request with a picture in a class will reveal `Learning` context for the student and `Working` context for the instructor. We use *isInContext* to associate context information to a post request.

2.1.2. Semantic Rules. There are two types of semantic rules: Inference rules (I) and Privacy rules (P). An agent uses inference rules to derive new information from the existing knowledge in its ontology. Privacy is mostly about subjective perceptions of the users. Hence, an agent should be aware of the privacy expectations of its user, which are represented with privacy rules. In a privacy rule, the user declares what type of post requests would be rejected at agreement time with other agents.

Semantic rules can be specified in two ways. (i) A user can use an interface to input her semantic rules. For example, Mester *et al.* implement their approach as a mobile application where the user inputs her privacy concerns in terms of ontological concepts [Mester et al. 2015]. (ii) Users mostly have difficulties to specify their privacy preferences [Sadeh et al. 2009]. Therefore, machine learning techniques can be investigated to automatically learn semantic rules of a user. Fang and LeFevre propose a

[1] We denote a `Concept` with text in mono-spaced format, a *property* with italic text, and an `:instance` with a colon followed by text in mono-spaced format.

Table I: Semantic Rules of Alice and Bob as SWRL Rules

I_{A_1}:	*foundAt*(?object, ?shop) → *isOrdinary*(?object, true)
I_{A_2}:	*hasUrl*(?shop, ?url1), *hasUrl*(?shop, ?url2), *differFrom*(?url1, ?url2) → *hasUrlBeside*(?shop, ?url1)
I_{B_1}:	*isInContext*(?postRequest, ?context), *hasMedium*(?postRequest, ?medium), *includesObject*(?medium, ?object), `Oktoberfest`(?location), *obtainedFrom*(?object, ?location), *isOrdinary*(?object, false) → `Festival`(?context)
I_{B_2}:	*hasOneUrl*(?shop, ?url), *isAccessible*(?url, false) → *isClosed*(?shop, true)
P_{B_1}:	`Festival`(?context), *isInContext*(?postRequest, ?context) → *rejects*(`:bob`, ?postRequest)

privacy wizard that automatically configures the users' privacy settings based on an active learning paradigm [Fang and LeFevre 2010]. Mugan *et al.* learn default personas that users can choose from to help users in specifying privacy settings in the domain of location sharing [Mugan et al. 2011].

In this work, we do not focus on how the semantic rules are specified. We assume that an agent is aware of the semantic rules of its user. We use Semantic Web Rule Language (SWRL) [Horrocks et al. 2004] to represent semantic rules. Each rule is of the form *Body* → *Head*, which means if the body holds then the head must also hold. The body and the head consist of conjunctions of atoms. Here, atoms are of the form `C`(x) and *P*(x,y). `C` is a concept name (e.g., `Festival`) and *P* is a property name (e.g., *isInContext*), which are defined in the ontology. `x` and `y` are either variables prefixed with a question mark (e.g., ?context), instance names (e.g., `:bob`) or literals (e.g., false). Semantic rules may depend on a specific location, context, relationship or any combination of these. The fact that an agent rejects a post request is represented by the use of *rejects*, which is the only property that can appear in the head of a privacy rule. Hence, all privacy rules are represented as singletons.

Table I shows the semantic rules of Alice and Bob. Alice has the inference rule I_{A_1}, which states that any object that can be found at a shop is ordinary; denoting that it is widely available to many people and not unique for an occasion. In addition to that, she has the inference rule I_{A_2}, which states that a shop has more than one website if the shop has two different urls. Bob has two inference rules. I_{B_1} states that if a post request has a medium that includes an object given at `Oktoberfest` then this post request is in `Festival` context. This is how Bob's agent can infer the context information from a given post request. I_{B_2} states that if the website of a shop is not accessible then that shop is out of business (denoted with `isClosed`). Bob has one privacy rule P_{B_1}, which states that any post request in `Festival` context should be rejected.

2.1.3. Reasoning. Each agent has an ontology that includes information about the social network domain, the content being shared by its user, the relationships and the privacy concerns of its user. While all agents share the concept descriptions of the social network domain, each agent has a set of instances in its ontology that might not be shared by all. That is, each agent understands the same thing from being a friend but Alice might not know who Bob's friends are. Agents automatically update their ontologies by following their users' actions (e.g., build a relationship) on the social networking site or by interacting with other agents.

When a user wants to share a post in a social networking site, she provides information about the post such as the audience, person or location tags and so on. Prior to posting, the agent (requesting agent) initiates a post request by using the post information. Then, it contacts all agents relevant to the post request to request permission. Those agents create a post request instance, update their ontology, and evaluate the post request according to the privacy concerns of their users. An agent deals with two types of information upon receiving a post request from a requesting agent: (i)

The agent can use the direct information provided in the post request. For example, the requesting agent may already know the users in the picture and can provide tag information of these users. Or the requesting agent may explicitly put the location information in a post request. (ii) The agent can use indirect information regarding the post request. For example, if the agent is equipped with face recognition methods, then it can automatically infer tags of the users appearing in the picture or the agent can infer the context information by analyzing past data of its user.

Here, we exemplify the use of indirect information through inference rules. An agent creates an ontological instance that represents the received post request. The agent uses its semantic rules to infer more information regarding the post request. Inference rules provide additional information about the post request (e.g., the context), which may be rejected in the case a privacy rule fires in the agent's ontology. So, the agent refers to its ontology to make a decision about accepting or rejecting a post request. If a privacy constraint of an agent is violated, those agents relevant to the post request provide arguments to make a collaborative decision. Agents can generate arguments from their own ontologies as well as by consulting other agents. We discuss our distributed argumentation approach in Section 4.

3. ARGUMENTATION FOR PRIVACY

Our work relies upon Assumption-based Argumentation (ABA) [Dung et al. 2009]. Formally, an ABA framework ($\mathcal{F}$) is a four-tuple $\langle\mathcal{L},\mathcal{R},\mathcal{A},\mathcal{C}\rangle$ with $\mathcal{L}$ the language, $\mathcal{R}$ a set of rules, $\mathcal{A}$ a set of assumptions and $\mathcal{C}$ a total mapping of contraries from $\mathcal{A}$ into $\mathcal{L}$. Each rule is of the form σ_1,...,$\sigma_m \rightarrow \sigma_0$ ($m \geq 0$, $\sigma_i \in \mathcal{L}$). The non-empty set of assumptions $\mathcal{A}$ is a subset of the language $\mathcal{L}$. An assumption is a weak point of an argument that can be attacked by another argument. So, each assumption has a contrary as defined in $\mathcal{C}$.

In ABA, an argument is of the form $S \vdash^R \sigma$, with $S \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$ and $\sigma \in \mathcal{L}$. S (the support) is a set of assumptions and σ (the claim) is derived using a set of rules R. A rule chain can be used to derive new conclusions as well; e.g., $R_3 = R_1 \cup R_2$ ($R_1, R_2, R_3 \in R$). In ABA frameworks, each assumption a is transformed into an argument of the form $\{a\} \vdash a$, which is supported by a and the empty set of rules. Each rule r with a body b and a head h is transformed into an argument such that the support contains all assumptions in b and the claim is h ($\{b.\text{assumptions}\} \vdash^r h$). An argument $S_2 \vdash \sigma_2$ is attacked by an argument $S_1 \vdash \sigma_1$ if and only if σ_1 is the contrary of one of the assumptions in S_2 [Dung et al. 2009; Toni 2014].

In today's social networks, when a user shares content, she shares it with her own privacy constraints. Here, we support that if the users can argue over the privacy constraints using ABA, then they can share it in a mutually agreed way. Thus, we would like to be able to describe privacy scenarios as ABA specifications. Table II shows how Scenario 5 can be expressed as an ABA specification with $\langle\mathcal{L},\mathcal{R},\mathcal{A},\mathcal{C}\rangle$. $\mathcal{L}$ is the language used in our ontology. $\mathcal{R}$ includes the rules that are defined in the ontologies of Alice and Bob as shown in Table I. Moreover, facts are also given as rules with empty bodies (r_1 - r_9). One such fact r_3 is that `:medium` includes a `:wband`. The assumption set $\mathcal{A}$ includes five assumptions of Alice and Bob. For example, one assumption (as_1) is that `:wband` can be found at `:Gifty`. $\mathcal{C}$ provides mappings between these assumptions and their contraries. For example, if `:alice` does not reject a post request `:pr` but `:bob` does, then these two sentences would be contradictory in the deductive system. Hence, these two sentences are defined as contraries.

The arguments that can be derived from such an instance are shown in Table III. Each fact f_i is supported by the empty set of assumptions and a rule r_i. All assumptions (a_1, a_2, b_1, b_4, b_6) are supported by the empty set of rules. The arguments a_3, a_4, b_2, b_3 and b_5 can be constructed given the rules in $\mathcal{R}$.

Table II: ABA Specification of Scenario 5 ($\mathcal{F}_1$)

$\mathcal{R} = I_{A_1} \cup I_{A_2} \cup I_{B_1} \cup I_{B_2} \cup P_{B_1} \cup_{i=1}^{9} r_i$
$r_1 = \{\rightarrow \textit{isInContext}(\texttt{:pr,:context})\}$
$r_2 = \{\rightarrow \textit{hasMedium}(\texttt{:pr,:medium})\}$
$r_3 = \{\rightarrow \textit{includesObject}(\texttt{:medium,:wband})\}$
$r_4 = \{\rightarrow \texttt{Oktoberfest(:location)}\}$
$r_5 = \{\rightarrow \textit{obtainedFrom}(\texttt{:wband,:location})\}$
$r_6 = \{\rightarrow \textit{taggedPerson}(\texttt{:medium,:bob})\}$
$r_7 = \{\rightarrow \textit{hasUrl}(\texttt{:Gifty,:url2})\}$
$r_8 = \{\rightarrow \textit{differFrom}(\texttt{:url,:url2})\}$
$r_9 = \{\rightarrow \textit{hasUrl}(\texttt{:Gifty,:url})\}$
$\mathcal{A} = \{as_1,as_2,as_3,as_4,as_5\}$
$as_1 = \textit{foundAt}(\texttt{:wband,:Gifty})$
$as_2 = \text{not}(\textit{rejects}(\texttt{:alice,:pr}))$
$as_3 = \textit{isOrdinary}(\texttt{:wband},\text{false})$
$as_4 = \textit{isAccessible}(\texttt{:url},\text{false})$
$as_5 = \textit{hasOneUrl}(\texttt{:Gifty,:url})$
$\mathcal{C} = \{c_1,c_2,c_3,c_4,c_5\}$
$c_1 = (\textit{foundAt}(\texttt{:wband,:Gifty}){=}\textit{isClosed}(\texttt{:Gifty},\text{true}))$
$c_2 = (\text{not}(\textit{rejects}(\texttt{:alice,:pr})){=}\textit{rejects}(\texttt{:bob,:pr}))$
$c_3 = (\textit{isOrdinary}(\texttt{:wband},\text{false}){=}\textit{isOrdinary}(\texttt{:wband},\text{true}))$
$c_4 = (\textit{isAccessible}(\texttt{:url},\text{false}){=}\textit{isAccessible}(\texttt{:url},\text{true}))$
$c_5 = (\textit{hasOneUrl}(\texttt{:Gifty,:url}){=}\textit{hasUrlBeside}(\texttt{:Gifty,:url}))$

Table III: Arguments derived from Scenario 5

$f_1 : \{\} \vdash^{r_1} \textit{isInContext}(\texttt{:pr,:context})$
$f_2 : \{\} \vdash^{r_2} \textit{hasMedium}(\texttt{:pr,:medium})$
$f_3 : \{\} \vdash^{r_3} \textit{includesObject}(\texttt{:medium,:wband})$
$f_4 : \{\} \vdash^{r_4} \texttt{Oktoberfest(:location)}$
$f_5 : \{\} \vdash^{r_5} \textit{obtainedFrom}(\texttt{:wband,:location})$
$f_6 : \{\} \vdash^{r_6} \textit{taggedPerson}(\texttt{:medium,:bob})$
$f_7 : \{\} \vdash^{r_7} \textit{hasUrl}(\texttt{:Gifty,:url2})$
$f_8 : \{\} \vdash^{r_8} \textit{differFrom}(\texttt{:url,:url2})$
$f_9 : \{\} \vdash^{r_9} \textit{hasUrl}(\texttt{:Gifty,:url})$
$a_1 : \{\textit{foundAt}(\texttt{:wband,:Gifty})\} \vdash \textit{foundAt}(\texttt{:wband,:Gifty})$
$a_2 : \{\text{not}(\textit{rejects}(\texttt{:alice,:pr}))\} \vdash \text{not}(\textit{rejects}(\texttt{:alice,:pr}))$
$a_3 : \{\textit{foundAt}(\texttt{:wband,:Gifty})\} \vdash^{I_{A_1}} \textit{isOrdinary}(\texttt{:wband},\text{true})$
$a_4 : \{\} \vdash^{I_{A_2} \cup_{i=7}^{9} r_i} \textit{hasUrlBeside}(\texttt{:Gifty,:url})$
$b_1 : \{\textit{isOrdinary}(\texttt{:wband},\text{false})\} \vdash \textit{isOrdinary}(\texttt{:wband}, \text{false})$
$b_2 : \{\textit{isOrdinary}(\texttt{:wband},\text{false})\} \vdash^{I_{B_1} \cup_{i=1}^{5} r_i} \texttt{Festival(:context)}$
$b_3 : \{\textit{isOrdinary}(\texttt{:wband},\text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i} \textit{rejects}(\texttt{:bob,:pr})$
$b_4 : \{\textit{isAccessible}(\texttt{:url},\text{false})\} \vdash \textit{isAccessible}(\texttt{:url},\text{false})$
$b_5 : \{\textit{hasOneUrl}(\texttt{:Gifty,:url}), \textit{isAccessible}(\texttt{:url},\text{false})\} \vdash^{I_{B_2}} \textit{isClosed}(\texttt{:Gifty},\text{true})$
$b_6 : \{\textit{hasOneUrl}(\texttt{:Gifty,:url})\} \vdash \textit{hasOneUrl}(\texttt{:Gifty,:url})$

The attacks between arguments are shown in Figure 1. For clarity, we omit the arguments (f_i) that cannot attack any arguments and cannot be attacked in the deductive system since they are supported by an empty set of assumptions. b_3 attacks a_2 because the claim of b_3 *rejects*(`:bob,:pr`) is defined as the contrary of not(*rejects*(`:alice,:pr`)), which is the support of a_2. On the other hand, a_3 attacks b_1, b_2 and b_3 since *isOrdinary*(`:wband`,true) is the contrary of *isOrdinary*(`:wband`,false), which is the support of b_1, b_2 and b_3. b_5 attacks a_1 and a_3 because *isClosed*(`:Gifty`,true) is the contrary of *foundAt*(`:wband,:Gifty`), which is the support of a_1 and a_3. At last, a_4 attacks b_5 because *ha-*

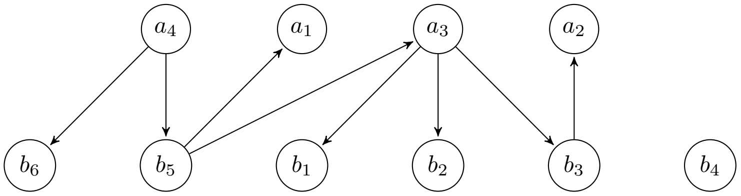

Fig. 1: Attacks between arguments for Scenario 5

sUrlBeside(`:Gifty`,`:url`) is the contrary of *hasOneUrl*(`:Gifty`,`:url`), which is the support of b_5 and b_6. There are no other attacks that can be derived from this ABA specification.

At this point, an agent can find the outcome of the discussion (i.e., if the post can be shared or not), since it has all of the arguments and the attack relations between these arguments. Agents benefit from ABA semantics and an ABA engine to calculate the winning arguments and to decide if to share the post.

Definition 3.1 (*Post sharing decision*). The requesting agent shares a post `:y` iff an argument $A \vdash$ *rejects*(`:x`,`:y`) is not valid in the argumentation framework, where $A \subseteq \mathcal{A}$ and `:x` is an agent relevant to `:y`.

Consider the argumentation between Alice and Bob. After the argumentation ends, Alice can use an ABA engine to check if her argument to share the picture is valid using a desired semantics. If so, according to Definition 3.1, the picture can be shared. In Figure 1, a_2 is attacked by b_3, which is attacked by a_3. Even though a_3 is attacked by b_5; a_4 defends a_3, which defends a_2. Under different semantics, this may lead to different outcomes. Assume admissible semantics is used [Dung 1995]. Then, a_2 is an admissible argument in $\mathcal{F}_1$ (see Table II). In other words, the claim not(*rejects*(`:alice`,`:pr`)) is acceptable (winning) for admissible semantics and thus, Alice shares the post.

4. DISTRIBUTED ARGUMENTATION

The previous section describes how argumentation would be used to argue for and against various privacy situations. That is, if Table II could be provided to an ABA engine, then the winning arguments could be computed. However, the contents of Table II are not available centrally. Furthermore, based on the content being shared as well as the users involved, the information such as the rules, assumptions, facts and contraries in the deductive system will change. In another words, agents will share relevant knowledge regarding the ongoing dialogue, choosing which information to share with whom depending on the context. This calls for a distributed generation of the system on demand (i.e., a dialogical framework).

We propose PRIARG (Privacy with Argumentation) to tackle privacy disputes in online social networks. The flow of PRIARG is depicted in Figure 2. When an agent (Agent A) wants to get consent from other agents to put up a post, it starts an argumentation session by sending an initial *case* (c) to the relevant agents. A case is a tuple of the form $\langle R, A, F, C, status\rangle$, with a set of rules R, a set of assumptions A, a set of facts F, a set of contraries C and a case $status$ that is $ongoing$ or $stop$. The receiving agent (e.g., Agent B) evaluates an $ongoing$ case and tries to extend it by attacking the set of assumptions. Hence, it adds the necessary rules, assumptions, facts, and contraries to the current case. Here, the agent can choose to use its central knowledge base (R_C+I_C), or it may consult other agents to collect rules and instances (R_D+I_D). Moreover, the agent can autonomously decide which subset of this information to share with other agents. If the received case cannot be extended, then the case status is set to $stop$. The exchange of cases is repeated sequentially until there is no contribution to the dispute. When the

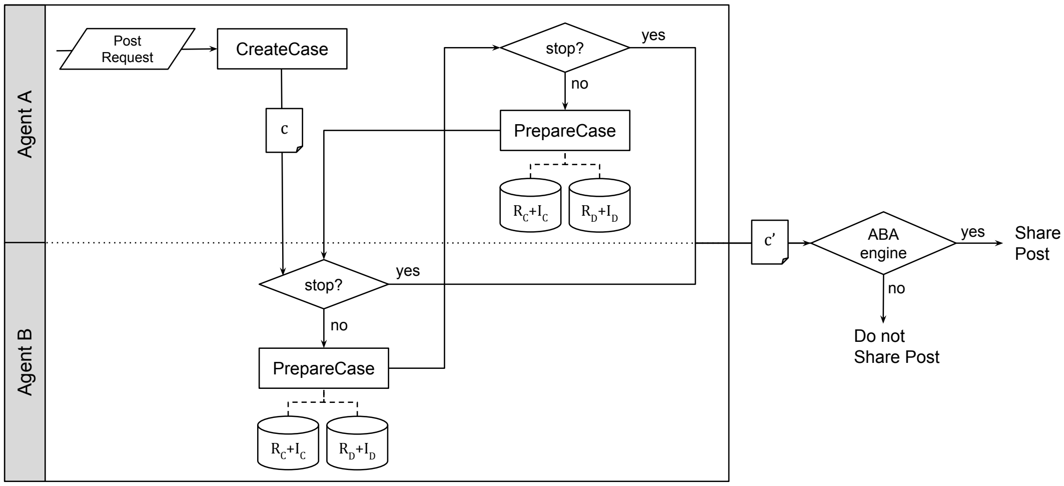

Fig. 2: Distributed Argumentation with PriArg between Agent A and Agent B

dispute stops, agents use the final case (c') to check whether the initial assumption to share the post is valid in the ABA engine. If it is valid, then the initial agent shares the post. Otherwise, the post is not shared; i.e., the other agent convinces the requesting agent not to share the post. ABA engine can be used internally by the agents as we do in this work. On the other hand, agents can invoke an external service (e.g., a web service that offers an ABA engine) to evaluate the final case.

4.1. Agent Profiles

Agents can exhibit different behaviors in providing information to a case since privacy concerns may change from one agent to another. We describe agent profiles that consider the cooperative behavior of heterogeneous agents. There are two important axis for describing agent profiles. The first axis is what information to consider when evaluating a post request. An agent can choose to use its internal knowledge, or it can enrich this knowledge with information coming from other agents. The second axis is how the information is disseminated. An agent can choose to share knowledge as it is, or it can apply some abstraction techniques to protect its privacy during information dissemination and only provide a more abstract information to a case.

4.1.1. Information Retrieval Types. Attacking the post request means that the agent can support a contrary from the previous case through rules, facts, and assumptions. An agent is free to use central or external information for decision making. Hence, we consider rules and instances (facts and assumptions) to be either centralized or decentralized. Here, we define four information retrieval types. An agent behaves according to the chosen type. Four possible types are as follows:

- *Type i* (Centralized Rules R_C + Centralized Instances I_C): In the most basic case, the receiving agent benefits from its ontology to find the necessary information to refute arguments of other agents. This corresponds to finding rules and their instances in the ontology. For example, assume that an agent is trying to show contrary *isOrdinary*(`:wband`, true). If it has I_{A_1} in its ontology (as in Table I) and the assumption *foundAt*(`:wband`,`:Gifty`), then it can easily show the desired contrary.
- *Type ii* (*Type i* + Decentralized Instances I_D): However, it is possible that even though the agent is aware of the rule, it does not have the necessary instances of the rule

body in its ontology (i.e., *foundAt*(`:wband`,`:Gifty`)). If this is the case, the agent will query other agents in its network to see if any of the agents can provide these instances. It is crucial to note that in a real application, one would expect the agent to consult those that it trusts for this particular task. For example, a shop owner would be more competent to provide this information than a young child. Our mechanism allows each agent to choose whom to consult for gathering information.

— *Type iii* (*Type i* + Decentralized Rules R_D): If the agent does not have rules that it can use to prove the contrary, it can then search for other agents who can provide relevant rules. This is similar to how people search for ways in solving problems. Again, one would need to either consult a trusted agent or consult many agents to see if there is an emerging consensus to decide to trust the rule. Once, the rule is trusted, the agent can use the assumptions or facts in its ontology to reach a conclusion. For example, the agent would consult others to find the rule I_{B_2} and then use a central instance (*hasOneUrl*(`:Gifty`,`:url`)).

— *Type iv* (*Type i* + R_D + I_D): Finally, the agent may find a trusted rule from others but may not have the necessary assumptions or facts to fire the rule. In this case, the agent would need to consult others for each of the predicates in the rule body and gather necessary assumptions and facts to fire the rule. Following the previous example, the agent can ask an agent for the predicates *isAccessible*(`:url`,false) and *hasOneUrl*(`:Gifty`,`:url`) to infer that $Gifty$ is out of business.

An agent can choose one of these types regarding its privacy concerns. Essentially, asking others for information can help an agent prepare a stronger attack because it might find new rules and instances.

4.1.2. Information Dissemination. During an argumentation, agents provide each other rules to explain the reasons behind an argument. However, revealing rules may also cause privacy violations. An agent can choose to hide some information from other agents. For this, an agent can share a more abstract information rather than sharing an exact one. For example, in Scenario 2, `:bob` rejects the post request of `:alice`, since the post is in the Festival context. Further, `:bob` states that the post is in the Festival context, since it includes a unique wristband that is taken from Oktoberfest. It is possible that `:bob` may not want `:alice` to learn this privacy concern. An abstraction mechanism can be used to prevent such situations. When an agent wants to reject a post request, it can generalize the rejection reasons instead of directly revealing them. For example, `:bob` can provide an abstracted rule to say that he does not want to share media (rather than pictures) in the Leisure (rather than Festival) context.

Agents can benefit from the hierarchy of classes and properties in their ontologies. To abstract a privacy rule, the agent obtains all of the predicates in the rule and searches for their ancestor predicates in its ontology (e.g., subsumed). In the ontology, `Festival` context is a subclass of the `Leisure` context, `Oktoberfest` is a subclass of `Location` and *includesObject* is a sub-property of *includes*. In Scenario 2, `:bob` can modify its exact rejection reason by abstracting its rule I_{B_1}, which would become: *isInContext*(?postRequest, ?context), *hasMedium*(?postRequest, ?medium), *includes*(?medium, ?object), `Location`(?location), *obtainedFrom*(?object, ?location), *isOrdinary*(?object, false) → `Leisure`(?context). Then, `:bob` updates the case with the abstract rule and instances. If the abstracted information is an assumption, then there is an interesting decision to make: whether to identify the contrary as the contrary of the original predicate or of the abstract one. Listing the contrary of the original predicate will help argumentation to yield a more accurate result, while listing the contrary of the abstract predicate will help preserve privacy. For example, if `:bob` would specify `Oktoberfest`(location) as an assumption and provide the abstract `Location`(?location), it could assign the contrary of `Oktoberfest`(location) as the contrary of `Location`(?location).

Since `:alice` would try to support the same contrary to continue the argumentation, abstraction would not change the result of the argumentation. Even though abstraction helps agents to conceal their privacy concerns, agents may not want to always use it or may prefer to use it based on whom the other agent is. PRIARG enables agents to decide whether to abstract privacy concerns and to what extend this abstraction should be made.

4.2. Cooperative Generation of a Case

We propose an algorithm, PREPAREATTACK, which can be used by an agent to attack post requests. Once the requesting agent shares the post request with other agents, each agent prepares attacks as described in Algorithm 1. This algorithm takes a case s as an input and returns an updated case s'. An argumentation session starts when an agent would like to share a post. At this point, this agent prepares a case by adding the post information to F, its assumption to share the post to A and the assumption contrary mappings to C. R is an empty set since there are no rules that the agent uses to attack an assumption. The case status is set to *ongoing* and sent to other agents with whom the argumentation will take place. Here, we assume that an agent sends cases to other agents who are mentioned or tagged in the post request. However, in principle, this can be personalized for each agent as needed. The following auxiliary functions are used in Algorithm 1:

— initCase() creates an empty case $\langle\{\},\{\},\{\},\{\},\text{-}\rangle$.
— updateOntology(F, $agent.ontology$) updates the agent's ontology in the case of having new facts, which contribute into the ontological reasoning of the agent.
— getContrariesToAttack(A, C) returns the contraries of assumptions that can be attacked.
— prepareCase(R, A, F, C, $status$) creates a case consisting of the rules R, the assumptions A, the facts F, the contraries C and *ongoing* or *stop* flag.
— getRelatedRules(c, o) finds the rules that can be used to infer c. This function is implemented differently based on an agent's profile. An agent can choose to use its own knowledge or consult other agents to find the relevant rules for proving c in its ontology. If others are consulted, then the agent asks for the predicate that needs to appear in the head of the rule. The returned rules need not be instantiated. The agent itself then finds the instances to instantiate the rule. Another aspect is the set of rules that the agent is willing to use. The agent may have a large set of rules that it can use to attack a contrary but can choose to use a few, based on its privacy concerns.
— getInstantiations(r, o) brings the instantiations of a rule. As the previous function, the semantics of this function changes according to the agent's profile. (i) If the rule (r) is fired then the rule body holds. In this case, the agent consults its own ontology (o) to instantiate the rule with its assumptions and facts. (ii) If the rule is not fired then there is not enough information for the rule body to hold. If the agent's profile allows the agent to collect information from others, then the agent can ask other agents to provide the necessary instances of the rule body. To do this, the agent sends the predicates in the body one by one to other agents to ask for instantiations. If the agent succeeds to collect these instances then this rule instantiation is considered as well.
— getBody(i) returns the predicates, which are part of the body of a rule instantiation i.
— getContrary(a) returns contrary of an assumption a.

PREPAREATTACK algorithm starts by initializing the response case s' (line 1). It checks whether the received case s is in a *stop* status (line 2) and if it is, s' becomes the received case s (line 24). Otherwise, the variables R, A, F and C are set to the rule set, the assumption set, the fact set and the contrary set as defined in s (line 3). The

ALGORITHM 1: PREPAREATTACK (s)

```
   Input: $s$, case received from other agent
   Output: $s'$, case sent to other agent
1  $s' \leftarrow$ initCase();
2  if $s.status \neq stop$ then
3      $R \leftarrow s.R$, $A \leftarrow s.A$, $F \leftarrow s.F$, $C \leftarrow s.C$;
4      $o \leftarrow$ updateOntology($F, agent.ontology$);
5      $contraryList \leftarrow$ getContrariesToAttack($A, C$);
6      foreach $c$ in $contraryList$ do
7          $rules \leftarrow$ getRelatedRules($c, o$);
8          foreach $r$ in $rules$ do
9              $iList \leftarrow$ getInstantiations($r, o$);
10             foreach $i$ in $iList$ do
11                 $R \leftarrow R \cup \{i\}$;
12                 foreach $p$ in getBody($i$) do
13                     if $p.name \in aList$ then
14                         $A \leftarrow A \cup \{p\}$;
15                         $p' \leftarrow$ getContrary($p$);
16                         $C \leftarrow C \cup \{p : p'\}$;
17                     else if $p.name \in fList$ : then
18                         $F \leftarrow F \cup \{p\}$;
19     if $R = s.R$ then
20         $s' \leftarrow$ prepareCase($R, A, F, C, stop$);
21     else
22         $s' \leftarrow$ prepareCase($R, A, F, C, ongoing$);
23 else
24     $s' \leftarrow s$;
25 return $s'$;
```

new facts are added to the agent's ontology and the agent infers further information given the new facts by the use of ontological reasoning (line 4). The agent finds the contraries $contraryList$ so that it can attack assumptions in A (line 5). The agent tries to support each contrary c in $contraryList$. For each contrary c, the agent finds a set of rules according to its agent profile (line 7). It is possible to have a rule with more than one instantiation. Each rule instantiation is a rule, where each variable in the rule is bound to an instance in the ontology. For each rule r, the agent gets all the instantiations of the rule according to its agent profile (line 9). For each rule instantiation i, i is added to R (line 11). Each predicate in the system is designated to serve as an assumption (in $aList$), a fact (in $fList$) or a deduction. This information comes from the domain ontology. The agent checks $aList$ and $fList$ to find the type of a predicate p in i's body. If p is an assumption, then it is added to A (line 14). In ABA, an assumption exists with its contrary, so agents also provide a contrary for each assumption. The assumption-contrary pair (p:p') is added to C (lines 15-16). If p is a fact, then it is added to F (line 18). If the agent cannot find any rules to attack an assumption, then R remains unchanged (line 19). Then, it creates a case s' in $stop$ status to indicate that the dispute should terminate (line 20). Otherwise, the dispute continues since the agent can attack at least one assumption in s and the agent prepares the case s' in $ongoing$ status by using the sets R, A, F and C (line 22). s' is returned by the algorithm (line 25). The complexity of the algorithm is bound by the number of contraries in the argumentation multiplied by the total number of rule instantiations used to attack the contraries. Since an agent can autonomously decide on the number of rules to use through getRelatedRules, the performance of the algorithm can be controlled by the agents themselves.

The sharing of a post will depend on how agents create and update cases. In this work, agents do not hide any information relevant to the ongoing dialogue (i.e., they do not use any abstraction technique). Moreover, agents are of *Type iv*. They implement getRelatedRules and getInstantiations methods in such a way that they consult other agents to collect all possible rules and instances for evaluating a case. That is, the agent could not have done anything more to improve the case regarding the discussion. Definition 4.1 captures the idea of a "complete" case. It provides a way to compare two cases for an agent based on how satisfactory the case is for that agent.

Definition 4.1 (*Complete Case*). Given a case $s = \langle R, A, F, C, status \rangle$ and any case $s' = \langle R', A', F', C', status' \rangle$ that are produced by an agent (w.r.t. a post request), s is a complete case iff $s' \subseteq s$; i.e., $R' \subseteq R$, $A' \subseteq A$, $F' \subseteq F$ and $C' \subseteq C$.

THEOREM 4.2. *Algorithm* PREPAREATTACK *always produces a complete case if agents are of Type iv.*

PROOF SKETCH. Let s be the complete case that could be produced by an agent. Assume that PREPAREATTACK produces s', which is not complete. This means that there exists a rule, assumption, fact or contrary that is in s but not in s' and that influences the outcome of the argumentation. However, PREPAREATTACK adds the influencing rules, facts, assumptions and contraries in lines (6-18). It uses the agent's ontology and the knowledge of other agents to prepare the case. Therefore, it produces the complete case s', that contradicts our initial assumption. □

4.3. To Share or Not to Share

When the privacy dispute terminates, agents can use the final case to make a decision. Note that since each case builds on top of the previous case, the final case contains all the necessary information (R, A, F and C) to form an ABA specification. This specification can then be provided to an ABA engine, which can generate arguments and attacks between arguments. Then, the requesting agent can use an ABA engine to decide to share the post or not (see Definition 3.1). Here, an important point is that the argumentation result can be different when different semantics are applied [Dung 1995]. Credulous semantics allows for alternative sets of arguments to be chosen as a winning argument set whereas skeptical semantics allows only one unique set of arguments to be acceptable. It depends on the application whether to use skeptical or credulous semantics to calculate the winning argument sets. If the resulting argument set is too critical and the application requires an uncontroversial argument set, it is better to use skeptical semantics. However, if the application requires at least one winning argument set under all circumstances, it is better to use credulous semantics [Toni 2014]. In this work, we use credulous semantics since we need to return a result to the agents under all circumstances. Further, we require admissible semantics. In admissible semantics, a set of arguments is admissible if it does not attack itself and it can defend itself against all arguments that attack it. In Figure 1, some alternative solutions are $\{a_4, a_3, a_2\}$, $\{a_3, a_2\}$, $\{a_4, a_3\}$ and so on. All these solutions are winning (acceptable) sets of arguments. However, there are other semantics in the literature such as preferred, complete, grounded and ideal. Here, an important point is that it is necessary for agents in an argumentation to use the same semantics to come to the same decision. Otherwise, one agent may find its sharing argument valid while the other agent finds it invalid.

Table IV: Cumulative iteration steps for Scenario 1 - Scenario 5

		Case					Rules		Instances		
Scen.	**Turn**	R	A	F	C	*status*	R_C	R_D	I_C	I_D	**Shared?**
1	`:alice`	$\{\}$	$\{\}$	$\cup_{i=1}^{6} r_i$	$\{\}$	-	-	-	$\cup_{i=1}^{6} r_i$	-	✓
2	`:alice`	$\{\}$	$\{as_2\}$	F	$\{c_2\}$	*ongoing*	-	-	as_2, F	-	-
	`:bob`	$\{I_{B_1},P_{B_1}\}$	$A\cup\{as_3\}$	F	$C\cup\{c_3\}$	*ongoing*	I_{B_1},P_{B_1}	-	as_3	-	-
	`:alice`	R	A	F	C	*stop*	-	-	-	-	✗
3	`:alice`	$R\cup\{I_{A_1}\}$	$A\cup\{as_1\}$	F	$C\cup\{c_1\}$	*ongoing*	I_{A_1}	-	-	as_1	-
	`:bob`	R	A	F	C	*stop*	-	-	-	-	-
	`:alice`	R	A	F	C	*stop*	-	-	-	-	✓
4	`:alice`	R	A	F	C	*ongoing*	-	-	-	-	-
	`:bob`	$R\cup\{I_{B_2}\}$	$A\cup\{as_4, as_5\}$	F	$C\cup\{c_4\}$	*ongoing*	-	I_{B_2}	-	as_4, as_5	-
	`:alice`	R	A	F	C	*stop*	-	-	-	-	✗
5	`:alice`	$R\cup\{I_{A_2}\}$	A	$F \cup_{i=7}^{9} r_i$	C	*ongoing*	I_{A_2}	-	$\cup_{i=7}^{9} r_i$	-	-
	`:bob`	R	A	F	C	*stop*	-	-	-	-	-
	`:alice`	R	A	F	C	*stop*	-	-	-	-	✓

5. EVALUATION

We have implemented PRIARG using Java and Spring frameworks. Agents communicate with each other using RESTful Web services. Each agent has three Web services: one to start the argumentation, one to consult other agents and the other to evaluate a post request and prepare an attack by using PREPAREATTACK algorithm. Agents use OWL API [Horridge and Bechhofer 2011] and the Pellet reasoner [Sirin et al. 2007] for evaluating SWRL rules and making ontological inferences. Each agent is equipped with an ABA engine *abagraph*[2] to compute the outcome of a dialogue. *abagraph* is an open source Prolog program, which is based on a graph-based argumentation derivation algorithm.

5.1. System Execution

Agents execute the five scenarios introduced in Section 2 as shown in Table IV. At each iteration, an agent receives a case and evaluates it. It uses its own knowledge (centralized rules R_C, centralized instances I_C) or consult others (decentralized rules R_D, decentralized instances I_D) to attack assumptions in the received case. It updates the case accordingly and sends a response case. Agents are able to see the result of the argumentation when the dispute terminates. For this, an agent puts a case, which involves all of the rules, assumptions, facts and contraries, into $abagraph$ and computes the outcome of the dialogue with credulous admissible semantics.

Scenario 1: Alice's agent (`:alice`) creates a case using her centralized instances $\cup_{i=1}^{6} r_i$. It puts all these instances to the set F. It does not want to consult `:bob` hence it does not provide any assumptions and contraries. `:alice` shares the post.

Scenario 2: `:alice` consults `:bob` before sharing the post. It creates a case such that A includes its assumption as_2, F all the facts, and C the contrary c_2. `:alice` uses her centralized instances to prepare the case. It sets the status of the case to $ongoing$ and sends it to `:bob`. `:bob` executes PREPAREATTACK upon receiving the case. `:bob` checks the status in the case and decides to continue argumentation. Then, it invokes updateOntology to update its ontology with the new facts so that more inferences can be made. The contrary *rejects*(`:bob`,`:pr`) holds in Bob's ontology. `:bob` uses its centralized rules and instances to prepare the case and it finds the related rules (I_{B_1} and P_{B_1}). While checking the predicates in the rule instantiations, it finds that *isOrdinary* is in $aList$, hence it updates the sets A and C. No more facts are added at this point. Since

[2] http://www.doc.ic.ac.uk/~rac101/proarg/abagraph.html

`:bob` added a new assumption as_3 to A and waits for an answer, it keeps the case status as $ongoing$. Then, it sends the case to `:alice`. `:alice` cannot attack any assumption by using its own ontology or collecting information from other agents. The dispute terminates and `:alice` does not share the post.

Scenario 3: `:alice` can attack only the assumption as_3 by supporting its contrary. Hence, it finds the related rule I_{A_1} in its own ontology and gets the instantiations that support *isOrdinary*(`:wband`, true). It asks another agent to provide the missing instance as_1. Then it adds I_{A_1} into R. It finds that *foundAt* is in $aList$ and updates the sets A and C. `:alice` sets the case status to *ongoing* and sends the case to `:bob`. `:bob` cannot prepare any attack, the dispute terminates and `:alice` shares the post.

Scenario 4: `:alice` sends the case $\langle R$,A,F,C,*ongoing*$\rangle$ to `:bob`. This time, `:bob` consults other agents to attack `:alice`'s post request. Another agent `:david` provides the rule I_{B_2}. `:bob` instantiates this rule with decentralized instances as_4 and as_5, which are also provided by `:david`. `:bob` updates the case and sends it to `:alice`. `:alice` terminates the dispute since it cannot support any of the contraries in C. `:alice` does not share the post.

Scenario 5: After `:bob` sends the case to `:alice`, `:alice` instantiates its centralized rule I_{A_2} with r_7, r_8 and r_9. Then, it updates the case and sends it to `:bob`. Since `:bob` cannot rebut the arguments of `:alice`, it sets the case status to *stop* and `:alice` shares the post. Note that this final case corresponds to the ABA specification described in Table II, which is generated in a distributed manner as a result of the argumentation.

5.2. Experimental Settings

We have conducted two survey-based experiments to understand what people expect to happen in various privacy scenarios. Then, we compare the solutions proposed by our framework and by the majority of people to understand if our framework provides solutions as expected by people. Both surveys consist of two parts. In the first part, we ask questions about the participant such as age, gender, frequency of social network usage. Moreover, we want to learn whether she has privacy concerns in social networks. In the second part, we present first four scenarios together with the privacy concerns of Alice and Bob (see Section 2). We ask questions to find out how participants would actually act in each scenario. For this, participants use the information provided to decide on whether it would be convenient to share the content at that particular situation. These two surveys are available online[3].

5.2.1. Personal Interviews. In this first study, we have directly worked with 36 respondents (9 females and 27 males) who were recruited from Department of Computer Engineering at Bogazici University. We have interviewed the participants in person and filled the questionnaire based on their answers. During this experiment, participants were warned several times to be objective while making a decision and to respect the information in the scenario rather than their personal privacy preferences. In this experiment, 33 (91.6%) respondents are graduate students aged from 18 to 35; the remaining respondents are instructors aged from 35 to 55. 30 (83.3%) respondents use social networking sites at least once a day. 28 (77.77%) respondents have privacy concerns in online social networks.

In the second part of the interview, for each scenario, we inform the participants of the privacy rules of Alice and Bob. We either use the word "know" or "think" when we give new information to emphasize the difference between a fact and an assumption.

[3] Our implementation code, the surveys and more example scenarios are available online on the project page http://mas.cmpe.boun.edu.tr/priarg.

For each scenario, participants are expected to make a decision about sharing the content or not. The personal interview results are shown in Table V. In these interviews, we had the opportunity to ask follow-up questions to understand the decision process of the participants.

Table V: Personal Interviews and Survey Results

Scenario	Personal Interviews (36 participants)		Online Survey (68 participants)	
	Share	Not Share	Share	Not Share
1	**83.33%**	16.66%	**64.71%**	35.29%
2	5.55%	**94.44%**	7.35%	**92.65%**
3	**52.77%**	47.22%	20.59%	**79.41%**
4	2.77%	**97.22%**	7.35%	**92.65%**

For the first scenario, 30 (83.33%) participants agreed on sharing the content as it is. Following the second scenario, there was a stronger consensus: 34 (94.44%) participants decided the picture not to be shared. Of the remaining two participants, one used her personal knowledge to claim that a wristband was not a strong evidence to relate a picture with a festival and the other emphasized that Bob's argument was not strong enough hence the picture could be shared. In the third scenario, the results were not this clear: 19 (52.77%) participants thought that Alice's argument was stronger than Bob's argument and thus the picture should be shared. However, the results show that participants had difficulties to make a decision in this scenario. In the fourth scenario, similar to Scenario 2, there was a strong consensus as 35 (97.22%) participants wanted the picture not to be shared. We observed that the participants felt more secure to suggest an item not to be shared to be on the safe side.

5.2.2. Online Survey. We have used QuestionPro[4] with the Academic License to create the second survey. The goal was to conduct a dialogue-based survey with more participants having different backgrounds. In the first part, we asked the general questions as before. We have disseminated the survey on Facebook. This survey was online for three days and it involved 68 participants (50 females and 18 males). 6 (8.82%) participants are aged between 18 and 25, 53 (77.94%) participants are aged between 25 and 45, and 9 (13.23%) participants are older than 45. 64 (94.12%) respondents use social networking sites at least once a day. 62 (91.18%) participants have privacy concerns in online social networks.

In the second part, we have presented the four scenarios as dialogues between Alice and Bob in a turn taking fashion. In each scenario, the participant was able to see the picture showing Bob with a wristband and the ongoing dialogue between Alice and Bob. For each scenario, the participant sees a dialogue and makes a decision about sharing the picture or not at that particular situation. Online survey results are shown in Table V.

The survey results are interesting. For Scenarios 2 and 4, we have a strong consensus not to share. In the second scenario, 63 (92.65%) participants agreed with Bob's argument and did not want to share the picture and in the fourth scenario, 63 (92.65%) participants found Bob's arguments acceptable and decided not to share the picture. These two scenarios are in agreement of our personal interviews. The result are somewhat different for the other two scenarios. In the first scenario, only 44 (64.71%) participants wanted the picture to be shared. This shows that the participants had a tendency of not sharing independent of the content since in this scenario there is no information about Bob's privacy being breached. In the third scenario, 54 (79.41%)

[4] http://www.questionpro.com

participants did not find Alice's argument strong enough and wanted to protect Bob's privacy by not sharing the picture. This is different than the results in personal interviews, but considering the fact that this group was already on the side of not sharing for Scenario 1, this is unsurprising.

5.2.3. Comparison with PRIARG *results.* One immediate observation is that both interviews and survey results show that subjects have tendency to change their decisions if they receive new information. Similarly, in PRIARG, agents provide arguments each time they confront new information. Each argument may change the result of the argumentation. This is an encouraging observation that shows that argumentation is indeed a good way to mimic how users think about privacy.

The second observation is that the results of PRIARG are mostly in line with the survey results. This is certainly the case for the two strong consensuses on Scenarios 2 and 4. PRIARG derives the "not share" result as almost all participants prefer. For Scenario 1, the survey results are not as strong as the results for Scenarios 2 and 4, but still the majority is in favor of sharing. PRIARG derives the "share" result as well. Scenario 3 is the most difficult to interpret because the two experiment results contradict with each other. In online survey, we can see that the participants have a bias towards not sharing the picture since 35.29% of them do not want to share the content in the first scenario. Then, this percentage increases in Scenario 3 as some of the participants wanting to share the picture in the first scenario change their decision. For this same scenario, the results of personal interviews do not say too much, half of the participants want to share the picture since they were warned to be objective while making a decision. For Scenario 3, PRIARG decides to share the content. That is, in our current setup, PRIARG is based on ABA using the credulous admissible semantics to find out winning arguments in a dispute. However, we notice that the participants who defend "not share" for Scenario 3 are in two groups.

The first group thinks that since Bob already has a privacy concern in Scenario 2, the system should respect that and not push any further unless the argument is very strong. We observed that this group has a tendency to not share the picture to make sure that nobody gets harmed. So that, they put more restrictions to the post than Bob normally does. This is akin to feeling that Bob is wronged and that the participants stand by him. In PRIARG this can be adjusted by choosing the right argumentation semantics. An agent that is concerned about privacy of other agents may prefer using skeptical semantics (rather than the credulous semantics) to make a decision. For example, in Scenario 3, the participants of the online survey are skeptical of sharing the picture since they want to protect Bob's privacy. Contrast this with an agent that takes the risk to violate the privacy of other agents. Such an agent may prefer using credulous semantics to make a decision. However, when the participants are warned as in the personal interviews to only use Bob's constraints and the provided information, the participants of personal interviews want to share the picture even if it may violate Bob's privacy.

The second group thinks that finding the wristband in a shop is not a strong argument. In order to handle such variations, the underlying argumentation scheme should support preferences or importance of arguments over others. ABA does not support this explicitly. However, different argumentation semantics that support preferences can be investigated. In ABA+ [Cyras and Toni 2016], agents can have preferences on assumptions. ABA+ uses preference information to reverse attacks in an argumentation session. In ASPIC+ [Modgil and Prakken 2013], preferences on rules are explicitly modeled to resolve conflicts between arguments. Value-based argumentation frameworks [Bench-Capon 2003] focus on the idea of arguments with values. Some attacks can be discarded according to the preference information [Bench-Capon

2003; Modgil and Prakken 2013]. In such a case, the argumentation can take into account the importance of argumentation elements (assumptions, rules, arguments) reflecting concerns of the users that we have seen in the interviews. The results of the argumentation would then be different. For example, in Scenario 3, if Bob's argument is stronger than Alice's argument then Alice cannot prepare an attack, the argumentation session would terminate and the picture would not be shared. This result would be in line with the result of online survey as well. Comparing different argumentation frameworks using various semantics could help mimic the concerns of various types of users. This is an interesting direction, which we leave as future work.

5.3. Comparative Evaluation

We evaluate our approach with respect to three leading approaches in the literature. Since there is neither a common data set, nor any previously reported results, we perform the comparison using desired properties for such systems. While there are many approaches for privacy, the approaches we choose are all aiming to facilitate agreement among privacy constraints.

PriNego is an agent-based system to resolve the privacy conflicts between users through negotiation [Mester et al. 2015]. The related agents provide reasons when they do not want a specific post to be shared and the agent who owns the post tries to modify the post or comes with an alternative one to satisfy the other agents' concerns.

CoPE is a privacy management system that runs as a Facebook application, where users create privacy policies for each shared image [Squicciarini et al. 2011]. The related users of a post are identified as co-owners. Each co-owner defines her own privacy settings to a post and the result is decided with voting.

FaceBlock uses an obscuring mechanism for pictures taken by Google Glass to protect the privacy of its users [Pappachan et al. 2014]. Users define their privacy rules with SWRL. FaceBlock uses a reasoner to check whether any privacy rule is triggered when a Google Glass device is detected. If so, FaceBlock sends the result of the privacy policy of the user to Google Glass device owner. If there exists a picture of the user taken by Google Glass and the user does not want to be seen, FaceBlock obscures the face of the user before sharing the picture.

Table VI: Comparison of Privacy Criteria

	PRIARG	CoPE	PriNego	FaceBlock
Automation	✓	✗	✓	✓
Concealment	✗	✓	✓	✓
Persuasion	✓	✗	✗	✗
External consultation	✓	✗	✗	✗

We use four criteria for comparing our approach to those in the literature (Table VI). The first two have been proposed by Mester *et al.* and the last two are new.

Automation refers to the ability of a system to work without human intervention. This mainly captures whether the approach is agent-based or whether the users themselves carry out the interactions with other users to reach an agreement. Our system is automated since there are agents, which represent users and act behalf of them. Similarly, each agent acts according to privacy rules of its user in PriNego. FaceBlock uses SWRL rules and a reasoner to act behalf of users. However, there is no automation in CoPE since users themselves try to build a shared privacy policy for each content.

Concealment refers to whether a system reveals the privacy rules of a user to other users. In PRIARG if a user decides to get involved in argumentation, she has to express her rules to show the reasons to support her arguments. This allows agents to provide meaningful support for each other's rules. For example, in Scenario 3, Alice gives information about a shop selling the wristband, because Bob has revealed his concern. Otherwise, there would not be a meaningful argumentation. However, using abstraction, an agent can also obscure the information it provides; thus, hiding its privacy constraint to some extent. In PriNego, agents provide reasons to each other but not share their privacy constraints. In FaceBlock, an agent sends the result of its privacy policy consisting of a directive about obscuring the face of its user. In CoPE, users do not provide rules. Hence, as opposed to the remaining approaches, the motivation for why the user does not want its picture to be shown is kept private.

Persuasion refers to whether a user in a system can question and rebut a claim of another user. PRIARG manages this by providing a platform for agents to discuss through arguments. PriNego attempts to enable this partially through negotiation. However, since the rules are concealed, the negotiation only allows agents to agree on a minimum common ground. CoPE or FaceBlock do not allow persuasion.

External consultation refers to whether users can benefit from other users' information to decide on how they will act to preserve their privacy. We consider this criterion an important one since people have a tendency to seek the knowledge of others to make a good decision. In our system, an agent can consult other agents to collect necessary rules or instances to refute an existing claim. This is important because sometimes an agent by itself may not know the implications of a situation itself but others around it can provide useful information to protect its privacy (see Scenario 4). The other three approaches do not have consultation with other agents.

6. DISCUSSION

We have proposed a framework where agents exchange arguments to decide whether a particular content will be shared or not. For this, each agent is equipped with an ontology that includes the social network domain information, the relationships of the agent, and the content. Moreover, privacy constraints of the user are kept as semantic rules in the ontology. Each agent uses its ontology to prepare a post request or evaluate a post request initiated by another agent. Agents can also consult other agents to collect information such as rules and instances. We have provided an algorithm that is used by the agents to exchange cases that encapsulate all necessary information to generate arguments. The final decision (share or not share) is made by an ABA engine. We have implemented a prototype, which relies on web services. Our user study shows that argumentation serves as a useful tool to mimic how humans deal with privacy disputes. Further, the solutions found by our framework are mostly in line with both interviews and survey results.

Argumentation has been used at various domains; from understanding micro-debates [Yaglikci and Torroni 2014] to improving education [Sklar and Parsons 2004]. Bentahar *et al.* use argumentation techniques to develop Business-to-Business (B2B) applications, where agents communicate with each other through abstract argumentation to resolve opinion conflicts [Bentahar et al. 2010]. Agents in the system have centralized rules and can use centralized or decentralized instances to generate an actual or partial argument. As opposed to our approach, their model does not support using decentralized rules. Their focus is on composition of web services whereas we focus on reaching a consensus on privacy.

Argumentation-based decision making models have been recently proposed in the literature, but they generally do not make use of ontologies. An exception is that of Williams and Hunter [Williams and Hunter 2007]. They combine argumentation with ontologies to develop a decision making system in treatment choice in breast cancer. Clinical trials are used to derive defeasible rules, which are then formulated in an ontology and arguments are created for ontological reasoning. However, they do not have a distributed approach as we advocate here, where the contents of the argumentation are generated on demand by the agents.

Some approaches focus on single agent decision making by using different argumentation frameworks. Amgoud and Prade make use of abstract argumentation for decision making under uncertainty [Amgoud and Prade 2009]. Müller and Hunter propose a model based on a simplified version of ASPIC+ [Muller and Hunter 2012]. Here, we use ABA for multiple agent decision making where an agent generates arguments by using its own ontology and consults other agents to reach a decision to share a post.

Fan *et al.* propose an approach to represent decision frameworks as ABA frameworks [Fan et al. 2014]. They show that admissible arguments found in ABA correspond to good decisions. They also focus on multiple agent decision making as we do here. Our work differs from theirs in the following aspects: (i) We use ontologies to represent knowledge, which enable agents to infer new information hence new arguments. (ii) Agents are involved in a persuasion dialogue and exchange their arguments in form of *cases* hence the number of utterances made are minimized. (iii) Agents can consult other agents to strengthen their arguments with new rules and instances.

Similar to the work of Mester *et al.* [Mester et al. 2015], Such and Rovatsos propose a mechanism to prevent privacy violations automatically by using a one-step negotiation protocol [Such and Rovatsos 2016]. Moreover, they propose heuristics to reduce the complexity of the negotiation mechanism. If negotiation techniques were to be applied in Scenario 4, then each agent would have evaluated the post request separately and exchange either concerns or alternative posts. With negotiation, Bob could have said that he is not happy with the context and ask Alice to put up a different picture. However, with argumentation, the agents can influence each other on their reasons. Similarly, if agents had employed voting, it would have been a tie between Alice and Bob and there would not be a clear winner. However, using PRIARG, the agents can reach an agreement.

Fogues *et al.* point out the need of a privacy recommendation tool to decide on the privacy settings of the posts to be published [Fogues et al. 2015]. They argue that such a tool should take the users' privacy constraints into consideration, and agents should negotiate these privacy constraints through arguments. Our motivations are aligned. However, we have also developed an algorithm for agents to engage in argumentation autonomously and decide on the result.

Wishart *et al.* propose an approach where users can define weak and strong privacy preferences regarding a particular item [Wishart et al. 2010]. According to these preferences, the system detects privacy violations so that the users can resolve the privacy conflicts manually. Our work is orthogonal to this in that we assume that the privacy rules are in place but provide a system to agree on the rules without human intervention.

Carminati and Ferrari propose a framework for specifying collaborative access control policies and enforcing such policies [Carminati and Ferrari 2011]. Hu *et al.* propose a model for multiparty access control for OSNs [Hu et al. 2013]. Similarly, Kökciyan and Yolum introduce a meta-model that enables agents to detect privacy violations in online social networks [Kökciyan and Yolum 2016]. The underlying idea in these works is that if the content access policies are specified correctly, then the system can detect whether there are violations. This is definitely useful. However, with our proposed approach, we are enabling agents to influence each other with additional information so that an agreement can be reached.

This work opens up interesting lines for research. Our current work does not question the trustworthiness of the information provided by the agents. That is, we have assumed that agents provide correct facts and rules. It would be interesting to adopt an epistemic perspective so that each agent can also have some (incomplete or wrong) beliefs about other agents. Hence, it becomes crucial for agents to monitor the results of their sharing and update their beliefs accordingly [Pilotti et al. 2015]. It would be also interesting to study how the algorithms would be affected if the information sources were trusted at different levels. An agent could systematically search for new information until it got nobody left to believe in. Another important direction is to incorporate preferences in dealing with privacy in the argumentation framework [Cyras and Toni 2016; Modgil and Prakken 2013; Bench-Capon 2003], so that certain privacy constraints are preferred to be preserved over others. The interview and survey results show that we should consider the importance of privacy rules reflecting privacy concerns of the users. Enabling preferences over rules and incorporating a way to reason on them could help establish this.

REFERENCES

Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. 2007. A unified and general framework for argumentation-based negotiation. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 158.

Leila Amgoud and Henri Prade. 2009. Using arguments for making and explaining decisions. *Artificial Intelligence* 173, 3 (2009), 413–436.

Trevor J. M. Bench-Capon. 2003. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation* 13, 3 (2003), 429–448.

Jamal Bentahar, Rafiul Alam, Zakaria Maamar, and Nanjangud C. Narendra. 2010. Using Argumentation to Model and Deploy Agent-based B2B Applications. *Knowledge-Based Systems* 23, 7 (2010), 677–692.

Barbara Carminati and Elena Ferrari. 2011. Collaborative access control in on-line social networks. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 231–240.

Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. 2011. Semantic web-based social network access control. *Computers & Security* 30, 2 (2011), 108–115.

Kristijonas Cyras and Francesca Toni. 2016. ABA+: assumption-based argumentation with preferences. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77, 2 (1995), 321–357.

Phan Minh Dung, Robert A Kowalski, and Francesca Toni. 2009. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*. Springer, 199–218.

Xiuyi Fan, Francesca Toni, Andrei Mocanu, and Matthew Williams. 2014. Dialogical Two-agent Decision Making with Assumption-based Argumentation. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 533–540.

Lujun Fang and Kristen LeFevre. 2010. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*. ACM, 351–360.

Ricard L. Fogues, Pradeep Murukanniah, Jose M. Such, Agustin Espinosa, Ana Garcia-Fornes, and Munindar Singh. 2015. Argumentation for multi-party privacy management. In *The Second International Workshop on Agents and CyberSecurity (ACySe)*. 3–6.

Fabien L Gandon and Norman M Sadeh. 2004. Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web* 1, 3 (2004), 241–260.

Matthew Horridge and Sean Bechhofer. 2011. The OWL API: A Java API for OWL Ontologies. *Semantic Web* 2, 1 (2011), 11–21.

Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, and others. 2004. SWRL: A semantic web rule language combining OWL and RuleML. *World Wide Web Consortium Member submission* 21 (2004), 79.

Hongxin Hu, Gail-Joon Ahn, and Jan Jorgensen. 2013. Multiparty access control for online social networks: model and mechanisms. *IEEE Transactions on Knowledge and Data Engineering* 25, 7 (2013), 1614–1627.

Nadin Kökciyan and Pınar Yolum. 2016. PriGuard: A Semantic Approach to Detect Privacy Violations in Online Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2724–2737.

Yavuz Mester, Nadin Kökciyan, and Pınar Yolum. 2015. Negotiating Privacy Constraints in Online Social Networks. In *Advances in Social Computing and Multiagent Systems*, Fernando Koch, Christian

Guttmann, and Didac Busquets (Eds.). Communications in Computer and Information Science, Vol. 541. Springer International Publishing, 112–129.

Sanjay Modgil and Henry Prakken. 2013. A general account of argumentation with preferences. *Artificial Intelligence* 195 (2013), 361 – 397.

Jonathan Mugan, Tarun Sharma, and Norman Sadeh. 2011. *Understandable learning of privacy preferences through default personas and suggestions*. Institute for Software Research Technical Report CMU-ISR-11-112. Carnegie Mellon University, Pittsburgh, PA.

Johannes Muller and Andrew Hunter. 2012. An argumentation-based approach for decision making. In *IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, Vol. 1. 564–571.

Primal Pappachan, Roberto Yus, Prajit Kumar Das, Tim Finin, Eduardo Mena, and Anupam Joshi. 2014. A Semantic Context-aware Privacy Model for Faceblock. In *Proceedings of the 2nd International Conference on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn)*. 64–72.

Pablo Pilotti, Ana Casali, and Carlos Iván Chesñevar. 2015. A Belief Revision Approach for Argumentation-Based Negotiation Agents. *Applied Mathematics and Computer Science* 25, 3 (2015), 455–470.

Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker, and Jinghai Rao. 2009. Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing* 13, 6 (2009), 401–412.

Albrecht Schmidt, Michael Beigl, and Hans-W Gellersen. 1999. There is more to context than location. *Computers & Graphics* 23, 6 (1999), 893–901.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web* 5, 2 (2007), 51–53.

Elizabeth Sklar and Simon Parsons. 2004. Towards the application of argumentation-based dialogues for education. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*. IEEE Computer Society, 1420–1421.

Anna C. Squicciarini, Federica Paci, and Smitha Sundareswaran. 2013. PriMa: a comprehensive approach to privacy protection in social network sites. *Annals of Telecommunications/Annales des Télécommunications* (2013), 1–16.

Anna C. Squicciarini, Heng Xu, and Xiaolong (Luke) Zhang. 2011. CoPE: Enabling Collaborative Privacy Management in Online Social Networks. *Journal of the American Society for Information Science and Technology* 62, 3 (2011), 521–534.

Margaret Gould Stewart. 2014. How giant websites design for you (and a billion others, too). (2014). Retrieved July 16, 2016 from https://goo.gl/vCO8IM

Jose M. Such and Michael Rovatsos. 2016. Privacy Policy Negotiation in Social Media. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 11, 1 (2016), 1–29.

Francesca Toni. 2014. A tutorial on assumption-based argumentation. *Argument & Computation* 5, 1 (2014), 89–117.

Douglas Walton and Erik C. W. Krabbe. 1995. *Commitment in Dialogue: Basic concept of interpersonal reasoning*. State University of New York Press, Albany NY.

Matt Williams and Anthony Hunter. 2007. Harnessing Ontologies for Argument-Based Decision-Making in Breast Cancer. In *IEEE International Conference on Tools with Artificial Intelligence*, Vol. 2. 254–261.

Ryan Wishart, Domenico Corapi, Srdjan Marinovic, and Morris Sloman. 2010. Collaborative Privacy Policy Authoring in a Social Networking Context. In *Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*. Washington, DC, USA, 1–8.

Nefise Yaglikci and Paolo Torroni. 2014. Microdebates App for Android: A Tool for Participating in Argumentative Online Debates Using a Handheld Device. In *26th International Conference on Tools with Artificial Intelligence (ICTAI)*. 792–799.