Argumentation vs Meta-argumentation for the Assessment of Multi-agent Conflict

Boris Galitsky[1], Sergei O. Kuznetsov[2] and Boris Kovalerchuk[3]

[1]Knowledge-Trail Inc. 9 Charles Str. Natick MA 01760, USA bgalitsky@searchspark.com
[2]Higher School of Economics, Moscow, Russia skuznetsov@hse.ru
[3]Dept. of Computer Science,Central Washington University, Ellensburg, WA, 98926 USA borisk@cwu.edu

Abstract

Comparative analysis of argumentation and meta-argumentation is conducted to track the plausibility of scenarios of interaction between agents. Meta-argumentation deals with the overall structure of a scenario, which included communicative actions in addition to attack relations and is learned from previous experience of multi-agent interactions. Object-level argumentation is a traditional machinery to handle argumentative structure of a dialogue, assessing the plausibility of individual claims. Evaluation of contribution of each argumentation level shows that both levels of argumentation are essential for assessment of multi-agent scenarios.

Introduction

Understanding and simulating behavior of human agents, as presented in text or other medium, is an important problem to be solved in a number of decision-making and decision support tasks (e.g. Fum et al 2007). Metareasoning has been identified as an efficient tool to simulate a cognitive side of human reasoning. In this study we propose a computational argumentation framework which is a two-level reasoning system on one hand, and is a solution to a practical problem where simulation of human behavior is required, on the other hand.

One of the approaches to simulation of human behavior is based on learning argument structures from previous experience with these agents, from previous scenarios of interaction between similar agents (Galitsky at al 2008). Another class of the solutions for this problem, based on the assessment of quality and consistency of argumentation of agents, has been attracting attention of the behavior simulation community as well (Chesñevar et al 2000).

In the context of agent-based decision support systems, the study of dynamics of *argumentation* (Prakken & Vreeswijk 2002) has proven to be a major feature for analyzing the course of interaction between conflicting agents (e.g. in argument-based negotiation or in multiagent dialogues (Weigand & de Moor 04). The issue of argumentation semantics of communicative models in the form of *attack relation* between facts (one fact defeats another) has also been addressed in the literature (eg Parsons et al 2002). Deductive meta-argumentation as proof of credulous acceptance of relationships between sets of arguments was introduced in (Doutre & Mangin 2004).

However, when there is a lack of background domain-dependent information, the evolution of *dialogues* ought to be taken into account in addition to the communicative actions these arguments are attached to. Rather than trying to determine the epistemic status of involved arguments, in one of our previous studies (Galitsky & Kuznetsov 08) we were concerned with the emerging *structure* of such dialogue argumentation. We refer to such structure as *meta-argumentation*. Meta-argumentation is implemented as a comparison with similar structures for other cases to mine for relevant ones for the purpose of assessing its truthfulness and exploration of a potential resolution strategy for multi-agent conflicts.

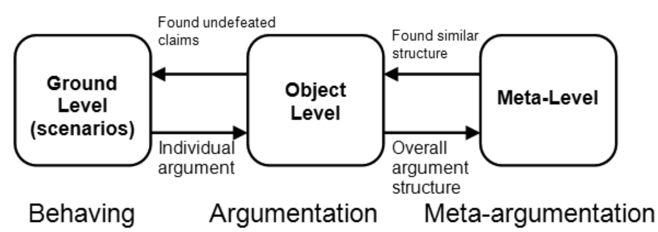


Figure 1. Relationship between behaving, argumentation and meta-argumentation.

Relationships between the ground level (multi-agent behavior), Object level (argumentation) and Meta-level (meta-argumentation) are shown in Fig.1 (The chart was proposed by (Cox & Raja 2007)). Two-level argumentation is a particular case of meta-cognition (Maheswaran & Szekely 2007), where *reasoning about* reasoning about plausibility of arguments is performed in a form of finding reasoning patterns of similar structure. Since object-level inference is *deduction* and meta-level inference is *induction*, although the meta-predicates range over object-level predicates, meta-level does not monitor or control object-level reasoning. Instead, if a fact can be attacked by other non-defeated facts, no meta-reasoning is required to observe that the scenario is invalid. If no such attack is found, meta-argumentation is used to verify if there are similar valid or invalid scenarios. Having found such similarity and having a particular fact defeated, meta-argumentation can give control to the object–level argumentation to operate with additional information. In this study we explore a simplified model where both levels of argumentation acts independently.

In our earlier studies we proposed a concept learning technique for scenario graphs, which encode information on the sequence of communicative actions, the subjects of communicative actions, the causal (Galitsky et al 05), and argumentation attack relationships between these subjects (Galitsky et al 08). Scenario knowledge representation and learning techniques were employed in such problems as predicting an outcome of international conflicts, assessment of an attitude of a security clearance candidate, mining emails for suspicious emotional profiles, and mining wireless location data for suspicious behavior (Galitsky et al 07).

In this study we perform a comparative analysis of the *two levels* of argumentation-related information mentioned above to assess plausibility of scenarios of interaction between agents. The *meta-level* of information on argumentation is an overall structure of a scenario, which included communicative actions in addition to attack relations and is learned from previous experience of multi-agent interactions. Scenarios are represented by directed graphs with labeled vertices (for communicative actions) and arcs (for temporal and causal relationships between these actions and their parameters) (Galitsky et al 05). The *object-level* is a traditional machinery to handle argumentative structure of a dialogue, assessing the plausibility of individual claims, which has been a subject of multiple applied and theoretical AI studies.

Meta-argumentation in dialogue

We approximate an *inter-human interaction scenario* as a sequence of communicative actions (such as *inform, agree, disagree, threaten, request*), ordered in time, with *attack* relation between some of the subjects of these communicative language. Scenarios are simplified to allow for effective matching by means of *graphs*. In such graphs, communicative actions and attack relations are the most important component to capture similarities between scenarios. Each vertex in the graph will correspond to a communicative action, which is performed by an (artificial) agent. As we are modeling dialogue situations for solving a conflict, we will borrow the terms *proponent* and *opponent* from dialectical argumentation theory (Prakken & Vreeswijk 2002) to denote such agents. An arc (oriented edge) denotes a sequence of two actions.

In our simplified model of communication semantics (Galitsky 2006) communicative actions will be characterized by three parameters: (1) *agent name*, (2) *subject* (information transmitted, an object described, etc.), and (3) *cause* (motivation, explanation, etc.) for this subject. When representing scenarios as graphs, we take into account all these parameters. Different arc types bear information whether the subject stays the same or not. **Thick arcs** link vertices that correspond to communicative actions with the **same subject**, whereas <u>thin arcs</u> link vertices that correspond to communicative actions with <u>different subjects</u>. We will make explicit conflict situations in which the cause of one communicative action M1 "attacks" the cause or subject of another communicative action M2 via an argumentation arc A (or *argumentation link)* between the vertices for these communicative actions. This attack relationship expresses that the cause of first communicative action ("from") *defeats* the subject or cause of the second communicative action ("to"). Such defeat relationship is *defeasible*, as it may be subject to other defeats, as we will see later.

A pair of vertices for a thick or thin arc may or may not be linked by the attack relation: a subject of the first communicative action is supported by a cause for the same (respectively, different) subjects of the second communicative action. However, we are concerned with argumentation arcs which link other than consecutive vertices (communicative actions) as shown at Fig. 2.

For the sake of example, consider the text given below representing a complaint scenario in which a client is presenting a complaint against a company because he was charged with an overdraft fee which he considers to be unfair. We denote both parties in this complaint scenario as ***Pro*** and ***Con*** (proponent and opponent), to make clear the dialectical setting. In this text communicative actions are shown in **bold**. Some expressions appear underline, indicating that they are defeating earlier statements. Fig. 3 shows the associated graph, where straight thick and thin arcs represent temporal sequence, and curve arcs denote defeat relationships.

Note that first two sentences (and the respective subgraph comprising two vertices) are about the current transaction (*deposit*), three sentences after (and the respective sub-graph comprising three vertices) address the *unfair charge,* and the last sentence is probably related to both issues above. Hence the vertices of two respective subgraphs are linked with thick arcs: *explain-confirm* and *remind-explain-disagree*. It must be remarked that the underlined expressions help identify where conflict among arguments arise. Thus, the company's claim <u>*as disclosed in my account information*</u> defeats the client's assertion *due to a bank error*. Similarly, the expression <u>*I made a deposit well in advance*</u> defeats *that it usually takes a day to process the deposit* (makes it non-applicable). The former defeat has the intuitive meaning "*existence of a rule or criterion of procedure attacks an associated claim of an* *error"*, and the latter defeat has the meaning "*the rule of procedure is not applicable to this particular case"*.

- *(Pro) I* ***explained*** *that I made a deposit, and then wrote a check which bounced due to a bank error.*
- *(Con) A customer service representative* ***confirmed*** *that it usually takes a day to process the deposit.*
- *(Pro) I* ***reminded*** *that I was unfairly charged an overdraft fee a month ago in a similar situation.*
- *(Con) They* ***explained*** *that the overdraft fee was due to insufficient funds as disclosed in my account information.*
- *(Pro) I* ***disagreed*** *with their fee because I made a deposit well in advance and wanted this fee back*
- *(Con) They* ***denied*** *responsibility saying that nothing can be done at this point and that I need to look into the account rules closer.*

Figure 2: A conflict scenario with attack relations.

Our task is to classify (for example, by determining its plausibility) a new complaint scenario without background knowledge, having a dataset of scenarios for each class. We intend to automate the above analysis given the formal representation of the graph (obtained from a user-company interaction in the real world, filled in by the user via a special form where communicative actions and argumentation links are specified).

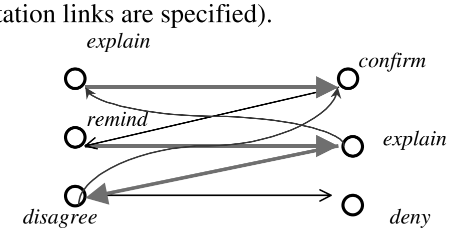


Figure 3: The graph for approximated scenario (Fig. 2).

Let us enumerate the constraints for the scenario graph:

1) All vertices are fully ordered by the temporal sequence (earlier-later);
2) Each vertex is either assigned with the proponent (drawn on the left side of each graph in Fig. 3) or to the opponent (drawn on the right side).
3) Vertices denote actions either of the proponent or of the opponent;
4) The arcs of the graph are oriented from earlier vertices to later ones;
5) Thin and thick arcs point from a vertex to the subsequent one in the temporal sequence (from the proponent to the opponent or vice versa);
6) Curly arcs, staying for *attack* relations, jump over several vertices in either direction.

Similarity between scenarios is defined by means of maximal common subscenarios. Since we describe scenarios by means of labeled graphs, we outline the definitions of labeled graphs and domination relation on them (see Kuznetsov 1999). Given ordered set G of graphs (V,E) with vertex- and edge-labels from the sets $(\Lambda_\varsigma, \preceq)$ and $(\Lambda_E, \preceq)$. A labeled graph Γ from G is a quadruple of the form $((V,l),(E,b))$, where V is a set of vertices, E is a set of edges, $l: V \to \Lambda_\varsigma$ is a function assigning labels to vertices, and $b: E \to \Lambda_E$ is a function assigning labels to edges.

The order is defined as follows: For two graphs $\Gamma_1:= ((V_1,l_1),(E_1,b_1))$ and $\Gamma_2:= ((V_2,l_2),(E_2,b_2))$ from G we say that Γ_1 *dominates* Γ_2 or $\Gamma_2 \le \Gamma_1$ (or Γ_2 is a *subgraph* of Γ_1) if there exists a one-to-one mapping $\varphi: V_2 \to V_1$ such that it

- respects edges: $(v,w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $l_2(v \preceq l_1(\varphi(v)),\ (v,w) \in E_2 \Rightarrow b_2(v,w) \preceq b_1(\varphi(v), \varphi(w))$.

This definition allows generalization ("weakening") of labels of matched vertices when passing from the "larger" graph G_1 to "smaller" graph G_2.

Now, generalization Z of a pair of scenario graphs X and Y (or their similarity), denoted by X ∗ Y = Z, is the set of all inclusion-maximal common subgraphs of X and Y, each of them satisfying the following additional conditions:

- To be matched, two vertices from graphs X and Y must denote communicative actions of the same agent;
- Each common subgraph from Z contains at least one thick arc.

The following conditions hold when a scenario graph U is assigned to a class:

1) U is similar to (has a nonempty common scenario subgraph of) a positive example R^+. It is possible that the same graph has also a nonempty common scenario subgraph with a negative example R^- . This is means that the graph is similar to both positive and negative examples.

2) For any negative example R^-, if U is similar to R^- (i.e., $U * R^- \neq \emptyset$) then $U * R^- \mu\ U * R^+$. This condition introduces the measure of similarity and says that to be assigned to a class, the similarity between the unknown graph U and the closest (in terms of μ) scenario from the positive class should be higher than the similarity between U and each negative example (i.e., representative of the class of implausible complaints).

Graph language is quite convenient to formalize the dialogues. However, to better show how the object level is related to the meta-level, we can represent both levels in a predicate language.

In the object level, we perform deduction with predicates *attackRelation(Fact1, Fact2)* to assess the validity of individual facts. Graph-based concept learning, expressed in predicate language, would include such predicates as

graph_edge(AttackRelation1(Fact11, Fact21), AttackRelation2((Fact21, Fact22)).

The scenario is then an ordered sequence of such expression, and a minimal common sub-scenario is the least general generalization operation of anti-unification. (Galitsky & Kuznetsov 2008) presents the metaprogramming technique for learning scenarios as logic programs, where the operation of anti-unification was adjusted to deal with conflict scenarios.

Using anti-unification requires associativity in finding similarity between scenarios**:** $(S_1 \cap S_2) \cap S_3 = S_1 \cap (S_2 \cap S_3)$. Applying anti-unification to scenarios as ordered lists of expressions for communicative actions does not obey the associativity. A naive solution here would be to ignore the order of communicative actions and just consider conjunctions of expressions for these actions. This would lead to ignorance of essential information about scenarios.

To overcome this problem we represent the ordered set of formulas for communicative actions as the unordered set of these action plus multiple instances of the binary predicate *after(Action1, Action2)* and *attackRelation(Fact1, Fact2)*, where *Fact1* and *Fact2* are the subjects of *Action1* and *Action2* respectively. Using this predicate allows retaining information about the order of communicative actions in scenario and obeying the associativity at the same time. To find a common scenario formula for two scenarios, which are represented by formulas above, we separately match predicates for communicative actions, predicates for their order of actions *after* and attack predicates. The role for scenario classification of a sequence of communicative actions forming a *step* is more important than the role of a single communicative action. Therefore, computing similarity between scenarios, one primarily needs to find matching steps occurring in a pair of scenarios and then search for matching individual actions.

Assessing defeasibility of individual claims

To verify the truthfulness of a complainant's claim, we use the special form called Interactive Argumentation Form which assists in structuring a complaint. Use of this form enforces a user to explicitly indicate all causal and argumentation links between statements which are included in a complaint. The form is used at the object-level argumentation to assess whether a particular scenario has plausible argumentation pattern: does it contain self-attacks (explicit for the complainant).

Beginning in July of 2003, I began using a non-Huntington account as my primary checking

I have 2 loans through Huntington, both of which were automatically deducted at the appropriate times each month from my Huntington account. At the beginning of July, I began paying those loans by check from the non-Huntington account. Though I had attempted to stop Huntington from taking the funds directly from my Huntington account, they continued to do so resulting in a continuing negative balance, compounded by NSF and overdraft fees, as well as the initial debit for the loans. Calls to Huntington regarding the matter have had no effect.

I'm constantly bombarded with calls from Huntington about this so called delinquency which culminated in a threat from Huntington collections to repossess my truck and other vehicle (both loan items)

When I explained that I had been paying the loans by check AND that those checks had been debited from my other bank account, they continued to insist that no payments had been applied to either account and that Huntington was still going to repossess my vehicles. Discussion with them caused me to look closer at my records and I've found corresponding checks that have posted from my primary, non-Huntington account.

It does appear, however, that one payment for $181.62 was never posted. After this, I again called Huntington and explained the situation. I was told that as long as Huntington had an open account for me, from which they'd already set up automatic withdraw, they could continue to withdraw funds for loan payment, even if the loan had already been paid by check! I was also informed that the best way to rectify the situation was to close the Huntington account.

Since getting my loan, I've had continuing trouble. The first payment was late, due to a mistake made by Huntington-which they acknowledged. Huntington told me that they'd take the late payment off my record but it appears they never did.

All in all, banking with Huntington has been a complete fiasco. I've been a loyal customer for over 21 years and expected to be treated with some amount of respect. Instead, I've been treated like a liar and made to feel that the responsibility in resolving this problem lies only on my shoulders.

Figure 4: Full complaint scenario

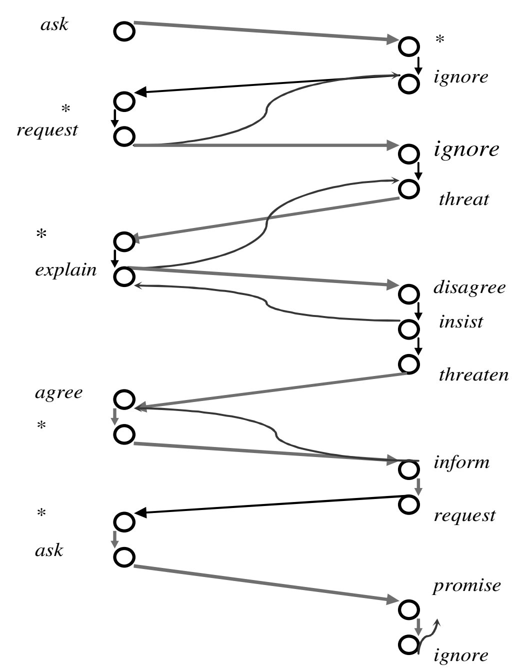


Figure 4a: Scenario graph for the above complaint (Fig. 4).

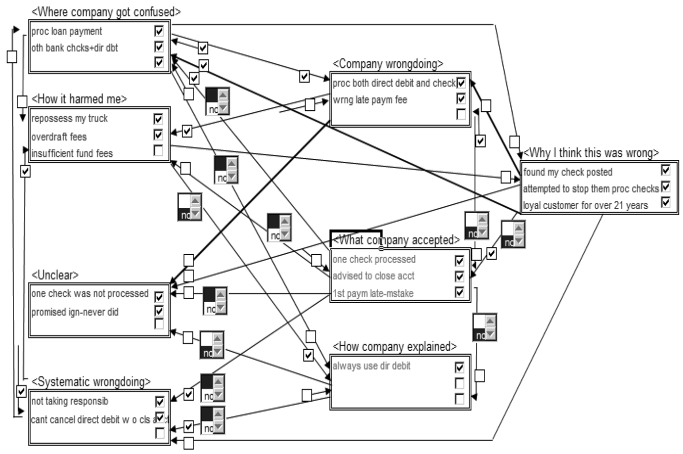
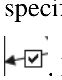
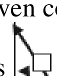


Figure 5: Interactive Argumentation Form

The form (Fig. 5) includes eight input areas where a complainant presents a component-based description of a problem (Fig.4). At the beginning, the subject of the dispute is specified: an operation (or a sequence of operations) which are believed by a complainant to be performed by a company in a different manner to what was expected <Where company got confused>. Then the essence of the problem is described, what exactly turned out to be wrong. In the section <Company wrongdoing> the complainant sketches the way the company performed its duties, which caused the current complaint. The customer's perception of the damage is inputted in section <How it harmed me>. In the fourth section <Why I think this was wrong> the customer backs up his beliefs concerning the above two sections, <Where company got confused> and <Company wrongdoing>.

All possible object-level argumentation links are shown as arrows. Arrows denote the links between the sentences in the respective sections; some arrows go one way and other both ways (only the ending portion is shown in this case). If the user does not find an arrow between two sections for a pair of inputted sentences, it means that either or both of these sentences belong to a wrong section: the data needs to be modified to obey the pre-defined structure. End of each arrow is assigned by a check-box to specify if the respective link is active for a given complaint . Bold arrows denote most important links .

The role of the Interactive Argumentation Form is a visual representation of argumentation, and intuitive preliminary analysis followed by the automated argumentation analysis. Since even for a typical complaint manual consideration of all argumentation links is rather hard, automated analysis of inter-connections between the complaint components is desired. We use the defeasible logic programming (García and Simari 2004) approach to verify whether the complainant's claims are plausible (cannot be defeated given the available data), concluding with the main claim, *Systematic wrongdoing*.

To specify supporting and defeating links for a number of statements for each section, multiple instances of these forms may be required for a given complaint. Since even for a typical complaint manual consideration of all argumentation links is rather hard, automated analysis of inter-connections between the complaint components is desired. We use the defeasible logic programming approach to verify whether the complainant's claims are plausible (cannot be defeated given the available data).

We enumerate the attack relations in the scenario

The fact that the first payment was late is accepted to be a mistake <What company accepted> → wrong late payment fee <Company wrongdoing>

The customer was charged the insufficient fund fee <How it harmed me> → it becomes clear how the bank makes money when customer claims that the bank always uses direct debit and no other way of loan payment is accepted <How company explained>.

The customer claims that the bank always uses direct debit <How company explained> → the bank cannot cancel direct debit without closing account <Systematic wrongdoing>

Making the mistake concerning charge for the first late payment <What company accepted> → not taking responsibility, in customer's opinion <Systematic wrongdoing>.

The customer perceives himself as a loyal one (for over 21 years) <Why I think this was wrong> → Making the mistake concerning charge for the first late payment <What company accepted>.

One processed check <What company accepted> → Processed both direct deposit and check of another bank <Company wrongdoing>.

Processed both direct deposit and check of another bank <Company wrongdoing> → less amount than expected in the checking account <commonsense> → negative balance in this account <commonsense> → insufficient fee <How it harmed me>.

Advised to close account <What company accepted> →other banks are able to cancel direct deposit orders <Where company got confused>

Dialectic trees for implicit self-attacks

In this section we provide the definition and algorithm for building dialectic trees to discover implicit self attack in a defeasible logic program, specified by the Interactive Argumentation Form (Figure 5).

Defeasible logic program (de.l.p.) is a set of facts, strict rules Π of the form (A:-B) , and a set of defeasible rules Δ of the form (A>-B). Let P=(Π, Δ) be a de.l.p. and L a ground literal. A *defeasible derivation* of L from P consists of a finite sequence $L_1, L_2, \ldots, L_n = L$ of ground literals, and each literal Li is in the sequence because:
(a) L_i is a fact in Π, or
(b) there exists a rule R_i in P (strict or defeasible) with head L_i and body $B_1, B_2, \ldots, B_k$ and every literal of the body is an element L_j of the sequence appearing before L_j ($j < i$).

Let h be a literal, and P=(Π, Δ) a de.l.p.. We say that $<A, h>$ is an *argument structure* for h, if A is a set of defeasible rules of Δ, such that:
1. there exists a defeasible derivation for h from =(Π ∪ A);
2. the set (Π ∪ A) is non-contradictory, and
3. A is minimal: there is no proper subset A_0 of A such that A_0 satisfies conditions (1) and (2).

Hence argument structure $<A, h>$ is a minimal non-contradictory set of defeasible rules, obtained from a defeasible derivation for a given literal h.

We say that $<A_1, h_1>$ *attacks* $<A_2, h_2>$ iff there exists a sub-argument $<A, h>$ of $<A_2, h_2>$ ($A \subseteq A_1$) so that h and h_1 are inconsistent. Argumentation line is a sequence of argument structures where each element in a sequence attacks its predecessor. There is a number of *acceptability* requirements for argumentation lines (Garcia & Simari 03).

We finally approach the definition of dialectic tree which gives us an algorithm to discover implicit self-attack relations in users' claims. Let $<A_0, h_0>$ be an argument structure from a program P. A *dialectical tree* for $<A_0, h_0>$ is defined as follows:
1. The root of the tree is labeled with $<A_0, h_0>$
2. Let N be a non-root vertex of the tree labeled $<A_n, h_n>$ and
$\Lambda = [<A_0, h_0>, <A_1, h_1>, \ldots, <A_n, h_n>]$ the sequence of labels of the
path from the root to N. Let $[<B_0, q_0>, <B_1, q_1>, \ldots, <B_k, q_k>]$ all attack
$<A_n, h_n>$. For each attacker $<B_i, q_i>$ with acceptable argumentation line $[\Lambda, <B_i, q_i>]$, we have an arc between N and its *child* N_i.

```
systematic_wrongdoing1(X) -< why_wrng1(X).
why_wrng1(X) -< how_it_harmed1(X).
how_it_harmed1('reposses my track').
~ why_wrng1(X). -< how_it_harmed1(X1),
company_accepted1 (X2).
company_accepted 1('one check processed').
~ why_wrng1(X) -< comp_confused1(X).
comp_confused1('proc loan payment').
~ unclear1(X)-< company_accepted2 (X1),
company_wrongdoing2(X2).
company_wrongdoing2(X) -< how_it_harmed2(X).
how_it_harmed2('overdraft fees').   ~ why_wrng1(X)-<
how_it_harmed1(X1), unclear1(X2).
unclear1(X)-< company_accepted2 (X).
company_accepted2 ('advised to close account').
company_accepted3 ('1st payment late - mistake').
~ unclear1(X)-<how_company_explained(X).
how_company_explained('always use direct debit').
~ company_wrongdoing2(X) -< company_accepted3 (X).
```

Figure 6: Defeasible logic program for a fragment of Interactive Argumentation Form on Fig. 5.

In a dialectical tree every vertex (except the root) represents an attack relation to its parent, and leaves correspond to non-attacked arguments. Each path from the root to a leaf corresponds to one different acceptable argumentation line. As shown in Fig.7, the dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is defeated. This tree is called dialectical because it represents an exhaustive dialectical analysis for the argument in its root, which is systematic wrongdoing of a company, the conclusion we want to draw or reject as a result of our analysis.

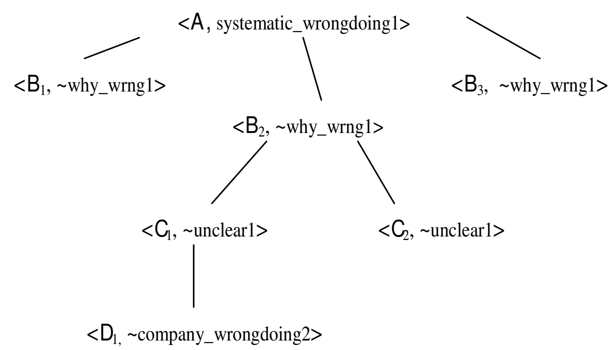


Figure 7: Dialectic tree for the Defeasible Logic Program Fig. 8.

For the defeasible logic program specified via the form, the dialectic tree shows that the main claim (the entry point) of the form is self-defeated, therefore complainants' claim of systematic company wrongdoing is not plausible.

Evaluation

To observe the comparative contribution of argumentation in object-level and meta-level, we used the database of textual complaints which were downloaded from the public website PlanetFeedback.com during three months starting from March 2004 and used in a number of computational studies since then. For the purpose of this evaluation, each complaint was:
1) manually assigned a plausibility assessment;
2) manually represented at a meta-level for machine learning evaluation;
3) manually represented as an object level for finding self-defeating claims.

We consider a complaint to be plausible if it seems to be consistent: no self-defeating is found, and implausible, if defeat is found (at either object-level or metalevel).

Not all complaints submitted by upset customers to consumer advocacy websites can be assessed with respect to plausibility. Clearly, the plausibility of those complaints just mentioning a failure in a product or a service without describing any interaction with the customer support cannot be assessed using the proposed technique.

This complaint preprocessing resulted in 560 complaints, divided in fourteen banks (or datasets), each of them involving 40 complaints. In each bank 20 complaints were used for training and 20 complaints for evaluation.

We performed the comparative analysis of relating scenarios to the classes of plausible/implausible taking into account 1), 2), and combined (1+2) . Such an analysis sheds a light on the possibility to recognize a scenario (1) without background knowledge, but reusing previous assigned argument structures, and (2) with partial background knowledge, expressed as a set of attack relations between claims. Furthermore, we evaluate a cautious approach combining 1) and 2), where scenario is plausible if a) it is similar to a plausible one **or** b) it does not contain self-defeated claims, and implausible otherwise.

Classification results are shown in Table1. On the left, the first three columns contain bank number, and the numbers of plausible/implausible complaints as manually assessed by human experts. The middle set of columns show the classifications results based on 1) & 2) together (separate 1) and 2) classification results are shown in appendix).

Bank	**As assigned by experts**		**Results of classification: both levels**								
	plausible	**implausible**	**Classified as plausible**	**implausible**	**plausible but implausible**	**implausible but implausible**	**Precision plausible**	**Precision implausible**	**Recall plausible**	**Recall implausible**	**F-measure**
1	**8**	**12**	8	11	0	1	100%	92%	100%	92%	96%
2	**6**	**14**	7	8	3	6	70%	57%	88%	57%	69%
3	**7**	**13**	9	11	2	2	82%	85%	82%	85%	83%
4	**5**	**15**	6	14	2	1	75%	93%	86%	93%	89%
5	**8**	**12**	8	10	0	2	100%	83%	100%	83%	91%
6	**8**	**12**	7	10	2	2	78%	83%	117%	83%	97%
7	**11**	**9**	11	9	0	0	100%	100%	100%	100%	100%
8	**8**	**12**	9	9	1	3	90%	75%	90%	75%	82%
9	**7**	**13**	7	11	0	2	100%	85%	100%	85%	92%
10	**9**	**11**	10	10	3	1	77%	91%	91%	91%	91%
11	**10**	**10**	10	8	0	2	100%	80%	100%	80%	89%
12	**5**	**15**	6	13	1	2	86%	87%	86%	87%	86%
13	**10**	**10**	10	9	0	1	100%	90%	100%	90%	95%
14	**8**	**12**	9	11	1	1	90%	92%	90%	92%	91%
avg	**7.86**	**12.1**	**8.4**	**10**	**1.071**	**1.857**	**89%**	**85%**	**95%**	**85%**	**89%**

Table 1. Results of the combined classification.

Classification based on the combination of levels gives substantial increase in recognition accuracy: F(1)= 63%, F(2) = 77%, and F(1+2)= 89%, which is a 26% of increase of accuracy for (1) and 12% increase of the accuracy for (2).

Results and discussions

In this study we observed how two levels of argumentation, overall argumentation pattern of a scenario and dialectic trees for individual claims, complement each other. Comparative computational analysis of scenario classification with respect to plausibility showed that both levels of argumentation (the former proposed in the current study, and the latter well known in various reasoning domains) are essential to determine whether a scenario is plausible or not (contains misrepresentation or self-contradiction). Hence we believe a practical argumentation management system should include scenario-oriented machine learning capability in addition to handling argumentation for individual claims.

Our approach can be related to metareasoning as a two-level system, however it is not a deductive metareasoning system. The object-level system is deductive, and the meta-level is inductive. Hence the proposed two-level system can be referred to as metareasoning in the same sense as induction is considered reasoning.

Graph-based concept learning benefits from argumentation information in the form of dialectic tree, because a representation graph G includes more sensitive data on how claims are attacking each other:

G = { *Communicative_actions* + *attack_relations_on_their_subject* + *vertices of dialectic tree*}.

Since sources for argumentation and meta-argumentation are different (although the nature of a self-defeat being discovered is similar), hybrid metareasoning system produces more accurate plausibility assessment. In terms of knowledge and meta-knowledge, argumentation and meta-argumentation gives a simple and clear example how these two sources of knowledge complement each other.

In our previous studies (Galitsky et al 2008) we verified that using attack relationship in addition to *Communicative_actions* as a way to express dialogue discourse indeed increases classification accuracy in a similar setting to the current study.

Dialectic trees work well when all relevant background knowledge is available, and has been represented in a form suitable for reasoning. Since it is never the case in practical application, argumentation leverages concept learning as an additional mechanism of acquiring data for individual claims from previous experiences with involved agents or types of agents. This sort of information would be unavailable without learning previous scenarios of multiagent interaction. Also, learning previous experience provides an online source for the preference on argument structures (acceptable argumentation lines require such preference).

We found an adequate application domain for computing dialectic trees such as assessment of plausibility of customer complaints. On one hand, this domain is a good source of experimental data for evaluation of argumentation structures because of a high volume of nontrivial scenarios of multiagent interaction, yielding a wide variety of de.l.ps. On the other hand, it is an important set of long-awaited features to be leveraged by customer relation management (CRM) systems.

We selected de.l.p. (García & Simari 03) as a most suitable approach to manage arguments in a dialogue, and employ dialectic trees to be integrated into scenario representation graph. (Dung 95) has proposed a very abstract and general argument-based framework, where he completely abstracts from the notions of argument and defeat. In contrast with (García & Simari 03) approach of defining an object language for representing knowledge and a concrete notion of argument and defeat, Dung's approach assumes the existence of a set of arguments ordered by a binary relation of defeat. In our case the source of this order is in the meta-level; it is the previous experience with involved agents.

Computational complexity associated with deductive reasoning is well known, especially for metareasoning. Hence using a limited knowledge at object level and machine learning instead of deduction at the meta-level assures applicability to practical problems.

We have not found any argumentation study concerned with matching the dialectic trees as a source of "global" structural information about scenarios. Meta-argumentation is usually used in deductive context (eg Hudelmaier 2008), but in this paper the level of reasoning about object-level argumentation is implemented as *induction*. A computational evaluation of how two *deductive* levels of argumentation complement each other is a subject of our further study.

References

Chesñevar, C., Maguitman, A. & Loui, R. Logical Models of Argument. ACM Computing Surveys 32(4) 337-383, 2000.

Chesñevar, C., Maguitman, A. An Argumentative Approach for Assessing Natural Language Usage based on the Web Corpus. In Proc. Of the ECAI 2004 Conf. Valencia, Spain, pp.581-585, 2004.

Cox, M., Raja, A. Metareasoning: a manifesto. BBN Technical Memo BBN TM-2028 www.mcox.org/Metareasoning/Manifesto/manifesto.pdf 2008.

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

Fum, D., Missiera, F. D. and Stoccob, A., The cognitive modeling of human behavior: Why a model is (sometimes) better than 10,000 words. Cognitive Systems Research v8 - 3, September 2007, 135-142.

Galitsky, B., Kovalerchuk, B. Kuznetsov, S.O. Learning Common Outcomes of Communicative Actions Represented by Labeled Graphs. ICCS , 387-400, 2007.

Galitsky, B. Reasoning about mental attitudes of complaining customers. Knowledge-Based Systems Elsevier. *Volume 19, Issue 7, 592-615* 2006.

Galitsky, B. Merging deductive and inductive reasoning for processing textual descriptions of inter-human conflicts. J Intelligent Info Systems v27 N1 21-48, 2006b.

Galitsky, B. & Kuznetsov, SO Learning communicative actions of conflicting human agents. JETAI 2008 (In the press).

García, A., Simari, G. Defeasible Logic Programming: an argumentative approach. *Theory and Practice of Logic Programming* 4(1), 95-138, 2004.

Hudelmaier J. Bicomplete calculi for intuitionistic propositional logic. www-ls.informatik.uni-tuebingen.de/hudelmaier/bicomp.pdf last downloaded Jan 3, 2008.

Kuznetsov S.O., Learning of Simple Conceptual Graphs from Positive and Negative Examples. In: J. Zytkow, J. Rauch (eds.), PKDD'99, LNAI vol. 1704 384-392, 1999.

Maheswaran, R.T. and Szekely, P. Metacognition for Multi-Agent Systems. In Metareasoning in Agent-Based Systems Workshop, Honolulu, HI, May 2007.

Prakken, H., Vreeswijk, G. Logical Systems for Defeasible Argumentation. In Gabbay, D. Guenther, F. Eds. Handbook of Phil. Logic, Kluwer, pp.219-318, 2002.

Parsons, S., Wooldridge, M., and Amgoud, L. An analysis of formal inter-agent dialogues, *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, Bologna 2002.

Weigand,H. & de Moor, A. Argumentation semantics of communicative action, Proceedings of the 9th Intl. Conference on the Language-Action Perspective on Communication Modelling (LAP 2004) Rutgers University, NJ, USA.

Doutre, S. & Mengin, J. On skeptical versus credulous acceptance for abstract argument systems. Logics In Artificial Intelligence: 9th European Conference, JELIA LNAI 3229 Alferes and Leite (Eds) 2004.