

Analyzing Conflicts with Concept-Based Learning

Boris A. Galitsky¹, Sergei O. Kuznetsov² and Mikhail V. Samokhin²

¹ School of Computer Science and Information Systems
Birkbeck College, University of London
Malet Street, London WC1E 7HX, UK
galitsky@dcs.bbk.ac.uk

² All-Russian Institute for Scientific and Technical Information (VINITI),
Usievicha 20, Moscow 125190, Russia
serge@viniti.ru, samohin_m@mtu-net.ru

Abstract. A machine learning technique for handling scenarios of interaction between conflicting agents is suggested. Scenarios are represented by directed graphs with labeled vertices (for mental actions) and arcs (for temporal and causal relationships between these actions and their parameters). The relation between mental actions and their descriptions gives rise to a concept lattice. Classification of an undetermined scenario is realized by comparing partial matchings of its graph with graphs of positive and negative examples. Developed scenario representation and comparative analysis techniques are applied to the classification of textual customer complaints.

1 Introduction: Reasoning with Conflict Scenarios

Scenarios of interaction between agents are an important subject of study in Artificial Intelligence. An extensive body of literature addresses the problem of logical simulation of agents' behavior, taking into account their beliefs, desires and intentions [1]. A substantial advancement has been achieved in building the scenarios of multiagent interaction, given properties of agent including their attitudes. Current approaches to the multiagent systems are either based on logical deduction [2,16] or simulation [4,14]; means of automated comparative analysis are still lacking [9].

In the former case, the sequence of mental states of agents is deduced from their initial mental states and initial attitudes. Deductive reasoning about actions and the logic with agents' attitudes as modalities are the most popular means to yield sequences of mental states of agents [20].

In the latter case, the system imitates the decision-making of agents, choosing the best action for each agent at each step, taking into account its current intentions, beliefs and desires, as well as those of others. Having the preference relation on the set of resultant states, each agent selects an action that is expected to lead to the most desired state [4].

However, a general framework to reuse the experience accumulated in previous scenarios of multiagent interaction has not been developed. For effective building and

predicting of interaction between agents, it is helpful to augment reasoning and/or simulation with machine learning [9,15,18,19]. It would reduce the number of possible agents' actions at each step, taking into account how these agents acted in previous cases.

Recently, the issue of providing BDI (Belief-Desire-Intention) agents [1] with machine learning capabilities attracted interest; an application domain such as agents for intelligent information access was considered in [8]. Nevertheless, a BDI-based machine learning framework for operating with scenarios of inter-human interactions was not developed yet. A number of case-based reasoning approaches have been suggested to treat the scenarios of interaction between BDI agents [15,18,19]; however, description of agents' attitude is reduced to their beliefs, desires and intentions in these studies. Indeed, behavior of real-world conflicting agents is described in a richer language using a wide number of mental entities including *pretending*, *deceiving*, *offending*, *forgiving*, *trust*, etc.

In this paper we build the representation machinery for conflict scenarios and propose a simple machine learning technique for classifying scenarios of multiagent conflict. This technique can be implemented in a stand-alone mode or used in combination with deductive reasoning or simulation.

Multiagent conflict is a special case of scenarios where the agents have inconsistent goals and a negotiation procedure is required to achieve a compromise [14]. In this paper we discover that following the logical structure of how negotiations are represented in text, it is possible to judge about consistency of this scenario [6].

Scenarios suggest the usage of complex data structure. In this paper we employ labeled directed acyclic graphs with arcs for describing interaction of two parties in a conflict, thus being within the standard concept graph representation [17]. A learning model needs to be focused on a specific graph representation for these conflicts. The learning strategies used here are based on ideas similar to that of Nearest Neighbors (see, e.g., [13]), case-based [10,12] and concept-based learning [7,11] or JSM-method [3]. Having defined scenarios and the operation of finding common subscenarios, we use the Nearest Neighbors and concept-based learning approach to relate a scenario to the class of valid or invalid scenarios.

The paper is organized as follows. The introduction of the domain of conflict scenarios is followed by a formal treatment of mental actions and defining a conflict scenario as a graph with vertices labeled by mental actions. Having defined the similarity operation on graphs (finding maximal common subgraphs), we present the procedure of relating a scenario to a class. The paper concludes with the description of the application domain of understanding customer complaints and the preliminary evaluation of the complaint data set.

2 The Domain of Conflict Scenarios

In this section we present our model of a conflict scenario oriented to the use in a machine learning setting. Here we develop a knowledge representation methodology based on approximation of a natural language description of a conflict [5].

When modeling scenarios of inter-human conflict, it is worth distinguishing mental and non-mental states and actions. The former include *knowing*, *pretending* (states) and *informing* or *asking* (actions); the latter are related, for example, to *location*, *energy* and *account balance* (physical states), as well as *moving*, *heating* and *withdrawal* (physical actions). It has been shown that an adequate description of mental world can be performed using mental entities and merging all other physical action into a constant predicate for an arbitrary physical action and its resultant physical state [5]. Furthermore, we express a totality of sequential mental states for a scenario via a set of mental actions that would unambiguously lead to these mental states. Hence we approximate an inter-human interaction scenario as a sequence of mental actions, ordered in time, with a causal relation between certain mental actions. Our approximation follows the style of situation calculus, scenarios are simplified to allow for effective matching by means of graphs.

Only mental actions remain as a most important component to express similarities between scenarios. Each vertex corresponds to a mental action, which is performed by either *proponent*, or *opponent*, the latter are called *agents* (here we consider two-agent systems, however, the model is easily extended to involve multiple agents). An arc (oriented edge) denotes a sequence of two actions.

In our model mental actions have two parameters: *agent name* and *subject* (information transmitted, a cause addressed, a reason explained, an object described, etc.). Representing scenarios as graphs, we take into account both parameters. Arc types bear information whether the subject stays the same. Thick arcs link vertices that correspond to mental actions with the same subject, thin arcs link vertices that correspond to mental actions with different subject.

The curve arcs denote a causal link between the arguments of mental actions, e.g., *service is not as advertised* \Rightarrow *there are particular failures in a service contract*, *ask* \rightsquigarrow *confirm*.

Let us consider an example of a scenario and its graph (Figure 1).

I explained that my cheque I wrote after I made a deposit bounced.
*A customer service representative **accepted** that it usually takes some time to process the deposit.*
I reminded that I was unfairly charged an overdraft fee a month ago in a similar situation.
*They **denied** that it was unfair because the overdraft fee was disclosed in my account information.*
*I **disagreed** with their fee and wanted it deposited back to my account.*
*They **explained** that nothing can be done at this point and that I need to look into the account rules closer.*

Note that first two sentences (and the respective subgraph comprising two vertices) are about the current transaction, three sentences after (and the respective subgraph 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-accept*) and (*remind-deny-disagree*).

In formal conflict scenarios extracted from text there can be multiple mental actions per step, for example *I disagreed ... and suggested....* The former mental action describes how an agent receives a message (*accept, agree, reject*, etc.) from an opponent, and the latter one describes the attitude of this agent initiating a request (*suggest, explain*, etc.), or reaction to the opponent's action. This division into *passive* (response) mode and *active* (request) mode is represented in the second attribute of mental actions specified in the second column of Table 2. Sometimes, either of the above actions is omitted in textual description of conflicts. Frequently, a mental action, which is assumed but not mentioned explicitly, can be deduced. In this paper for the sake of simplicity we will consider single action per step, performing the comparative analysis of scenarios.

There is a commonsense causal link *between being charged an unfair fee and intention to have this amount of money back* which is expressed by the arc between *remind* and *disagree*. Semantically, arcs with causal labels between vertices for mental actions express the causal links between the arguments of mental actions rather than between the mental actions themselves.

In our further analysis we will show how to relate this scenario (denoted as U) to the class of negative (unjustified) complaints.

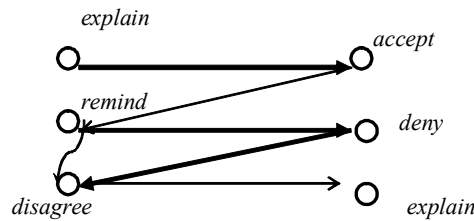


Fig. 1. The graph for approximated scenario

3 Semantics of Mental Actions

As to the choice of mental actions to adequately represent multiagent conflicts, we have selected the most frequently used from our structured database of complaints (Table 1).

To express the similarity between mental actions, we introduce five attributes each of which reflects a particular semantic parameter for mental activity (Table 2):

- *Positive/negative attitude* expresses whether a mental action is a cooperative (friendly, helpful) move (1), uncooperative (unfriendly, unhelpful) move (-1), neither or both (hard to tell, 0).
- *Request / respond mode* specifies whether a mental action is expected to be followed by a reaction (1), constitutes a response (follows) a previous request, neither or both (hard to tell, 0).

- *Info supply / no info supply* tells if a mental action brings in an additional data about the conflict (1), does not bring any information (-1), 0; does not occur here.
- *High / low confidence* specifies the confidence of the preceding mental state so that a particular mental action is chosen, high knowledge/confidence (1), lack of knowledge/confidence (-1), neither or both is possible (0).

Intense / relaxed mode says about the potential emotional load: high (1), low (-1), neutral (0) emotional loads are possible.

Table 1. The set of mental actions from a typical complaint

Customer describes actions of himself	Customer describes actions of the Company
Agree, explain, suggest, bring company's attention, remind, allow, try, request, understand, inform, confirm ask, check, ignore, convince disagree, appeal, deny, threaten	Agree, explain, suggest, remind, allow, try, request, understand, inform, confirm, ask, check, ignore, convince, disagree, appeal, deny, threaten, bring to customer's attention, accept complaint, accept /deny responsibilities, encourage, cheat

Table 2. Many-valued context of mental actions

Mental action	Attributes				
	Positive/negative attitude	Request / respond mode	Info supply / no info supply	High / low confidence	Intense / relaxed mode
agree	1	-1	-1	1	-1
accept	1	-1	-1	1	1
explain	0	-1	1	1	-1
suggest	1	0	1	-1	-1
bring_attent	1	1	1	1	1
remind	-1	0	1	1	1
allow	1	-1	-1	-1	-1
try	1	0	-1	-1	-1
request	0	1	-1	1	1
understand	0	-1	-1	1	-1
inform	0	0	1	1	-1
confirm	0	-1	1	1	1
ask	0	1	-1	-1	-1
check	-1	1	-1	-1	1
ignore	-1	-1	-1	-1	1
convince	0	1	1	1	-1
disagree	-1	-1	-1	1	-1
appeal	-1	1	1	1	1
deny	-1	-1	-1	1	1
threaten	-1	1	-1	1	1

Note that out of the set of meanings for each mental action (entity, speech act), we merge its subset into a single meaning, taking into account its relations to the mean-

ings of other mental actions [5]. Our approach follows along the lines of the theory of speech acts [23, 25] in its ability to handle performatives [22]. The theory of performatives is proposed as a test case for the rationality-based theory of mental actions such as *threatening*, *warning*, or *promising* that are carried out simply by saying the appropriate words. It is shown how "I request you..." can be a request, and "I lie to you that ..." can be self-defeating. The analysis [22] supports and extends the systematic account [24] of the roles of the speaker's communicative intention and the hearer's inference in literal, nonliteral and indirect uses of sentences to perform speech acts. Table 2 is obtained as a reduction of speech act's attributes to the case of multiagent conflicts.

An alternative way to express the set of selected meanings for each mental action uses an expression in the *want-know-believe* basis (Figure 2), presented in [1,20], extending the BDI model [4,5]. Note that clauses may be embedded as arguments for mental actions as meta-predicates. We refer the reader to [5] for further details on defining mental actions and mental states in the above basis. For example, various meanings of mental action *inform* are expressed as follows:

```
inform(Who, Whom, What) :- want(Who, know(Whom, What)),
    believe(Who, not know(Whom, What)),
    believe(Who, want(Whom, know(Whom, What))).
% The most general definition
inform(Who, Whom, What) :- believe(Who, know(Whom, What)),
    want(Who, believe(Whom, know(Who, What))).
%to inform Whom that not only Whom but Who knows What
inform(Who, Whom, What) :- ask(Whom, Who, What),
    want(Who, know(Whom, What)).
% informing as answering
inform(Who, Whom, What) :- ask(SomeOne, Who, believe(Whom, What)),
    want(Who, know(Whom, What)).% following SomeOne's request for informing
```

```
disagree(A,B,W) :- inform(A,B,W), not believe(B,W), inform(B,A, not W).
agree(A,B, W) :- inform(A,B, W), believe(B, W), inform(B,A, W).
explain(A,B, W) :- believe(A, (W :- V)), not know(B, W), inform(A,B,V),
    inform(A,B,(W :- V)), believe(B,W).
confirm(A,B, W) :- inform(A,B,W), know(A, believe(B, W)).
bring_attention(A,B, W) :- want(A, believe(B, know(A, W))).
remind(A,B, W):- believe(A, believe(B, W)),
    inform(A,B,W), want(A, know(B, know(A, W))).
understand(A,W) :- inform(B,A,W), believe(B, not believe(A, (W :- V))),
    want(B, believe(A, (W :- V))), inform(B, A,(W :- V)), believe(A,(W :- V),
    believe(A, W).
acceptResp(A, W) :- want(B, not W), believe(B, (W:-do(A,W1))),
    want(A, know(B, believe(A, (W:-do(A,W1))))), inform(A,B, (W:-do(A,W1))).
```

Fig. 2. The clauses for the selected mental entities from Table 2

To represent the hierarchy of mental actions by a concept lattice, we scale nominally the first and second attributes (i.e., the attribute values -1, 0, and 1 are considered as completely dissimilar). The third, fourth, and fifth attributes are already two-valued. Thus, the scaled context has seven attributes and the resulting concept lattice is presented in Figure 3. *ConExp* [21] software was used to construct and visualize the concept lattice [8] of mental actions and their attributes.

The concept lattice illustrates the semantics of mental actions; it shows how the choice of natural language semantics for mental entities covers the totality of meanings in the knowledge domain of interaction between agents.

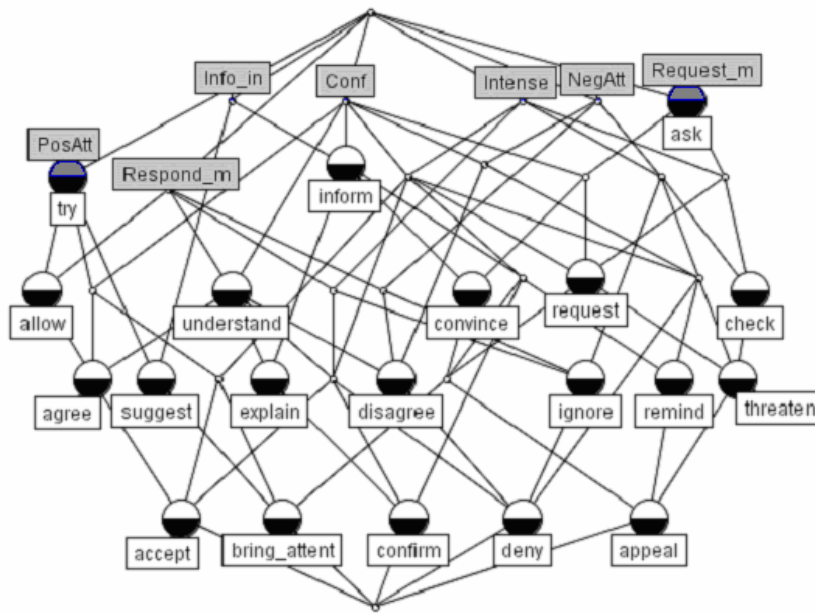


Fig. 3. The concept lattice for mental actions

4 Defining Scenarios as Graphs

We proceed with the description of our scenario dataset. This dataset contains two sets of complaint scenarios: showing a good attitude of a complainant (consistent plot with proper argumentation, a *valid complaint*) on the left, and a bad attitude of a complainant (inconsistent plot with certain flaws, implausible or irrational scenarios, an *invalid complaint*) on the right (Figure 4).

Each scenario includes 2-6 interaction *steps*, each consisting of mental actions with the alternating first attribute $\{request - respond - additional request or other follow up\}$. A step comprises one or more consequent actions with the same subject. Within a step, vertices for mental actions with common argument are linked with *thick arcs*.

For example, *suggest* from scenario V2 (Figure 4) is linked by a thin arc to mental action *ignore*, whose argument is not logically linked to the argument of *suggest* (the subject of suggestion). The first step of V2 includes *ignore-deny-ignore-threaten*; these mental actions have the same subject (it is not specified in the graph of conflict scenario). The vertices of these mental actions with the same argument are linked by the *thick arcs*. For example, it could be **ignored** *refund because of a wrong mailing address*, **deny** *the reason that the refund has been ignored [because of a wrong mailing address]*, **ignore** *the denial [...concerning a wrong mailing address]*, and **threatening** *for that ignorant behavior [...concerning a wrong mailing address]*. We have *wrong mailing address* as the common subject S of mental actions *ignore-deny-ignore-threaten* which we approximate as

$ignore(A1, S) \& deny(A2, S) \& ignore(A1, S) \& threaten(A2, S)$, keeping in mind the scenario graph. In such approximation we write $deny(A2, S)$ for the fact that $A2$ denied the reason that the refund has been ignored because of S . Indeed, $ignore(A1, S) \& deny(A2, S) \& ignore(A1, S) \& threaten(A2, S)$. Without a scenario graph, the best representation of the above in our language would be

$ignore(A1, S) \& deny(A2, ignore(A1, S)) \& ignore(A1, deny(A2, ignore(A1, S))) \& threaten(A2, ignore(A1, deny(A2, ignore(A1, S))))$.

Let us enumerate the constraints for the scenario graph:

- 1) All vertices are fully ordered by the temporal sequence (earlier-later);
- 2) Each vertex has a special label relating it either to the proponent (drawn on the right side in Figure 4) or to the opponent (drawn on the left 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 causal links, can jump over several vertices.

Similarity between scenarios is defined by means of maximal common subscenarios. Since we describe scenarios by means of labeled graphs, first we consider formal definitions of labeled graphs and domination relation on them (see, e.g., [7]).

Given ordered set G of graphs (V, E) with vertex- and edge-labels from the sets (\mathcal{L}_V, \preceq) and (\mathcal{L}_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, $l: V \rightarrow \mathcal{L}_V$ is a function assigning labels to vertices, and $b: E \rightarrow \mathcal{L}_E$ is a function assigning labels to edges. We do not distinguish isomorphic graphs with identical labelings.

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 $\Gamma_2 \leq \Gamma_1$ (or Γ_2 is a **subgraph** of Γ_1) if there exists a one-to-one mapping $\varphi: V_2 \rightarrow V_1$ such that it

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

Note that 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 $X \sqcap Y = Z$, 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 mental actions of the same agent;
- Each common subgraph from Z contains at least one thick arc.

This definition is easily extended to finding generalizations of several graphs (e.g., see [7, 11]). The subsumption order \sqsubseteq on pairs of graph sets X and Y is naturally defined as $X \sqsubseteq Y := X \cap Y = X$.

After scaling the many-valued context of mental actions, descriptions of mental action are given by 9-tuples of attributes, ordered in the usual way. Thus, vertex labels of generalizations of scenario graphs are given by intents of the scaled context of mental actions (see Figure 3).

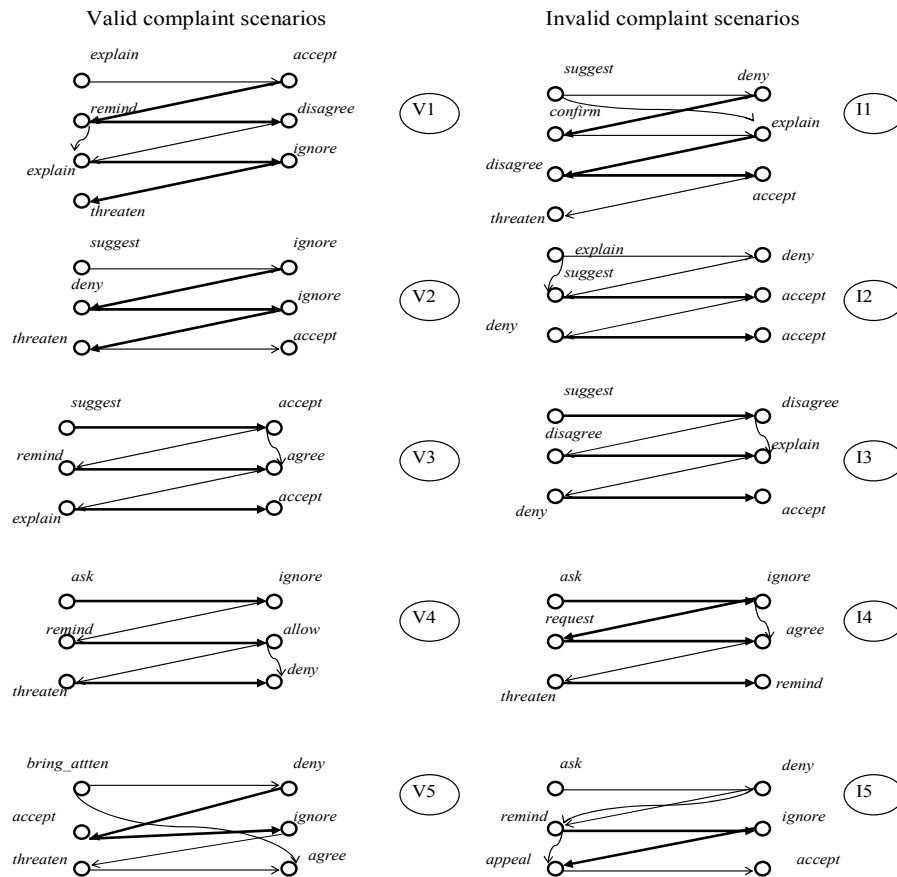


Fig. 4. The training set of scenarios

If the conditions above cannot be met then the common subgraph does not exist.

5 Relating a Scenario to a Class

Here we propose two schemes for classifying scenarios, given examples from positive and negative classes (see an example of a training sample in Figure 4).

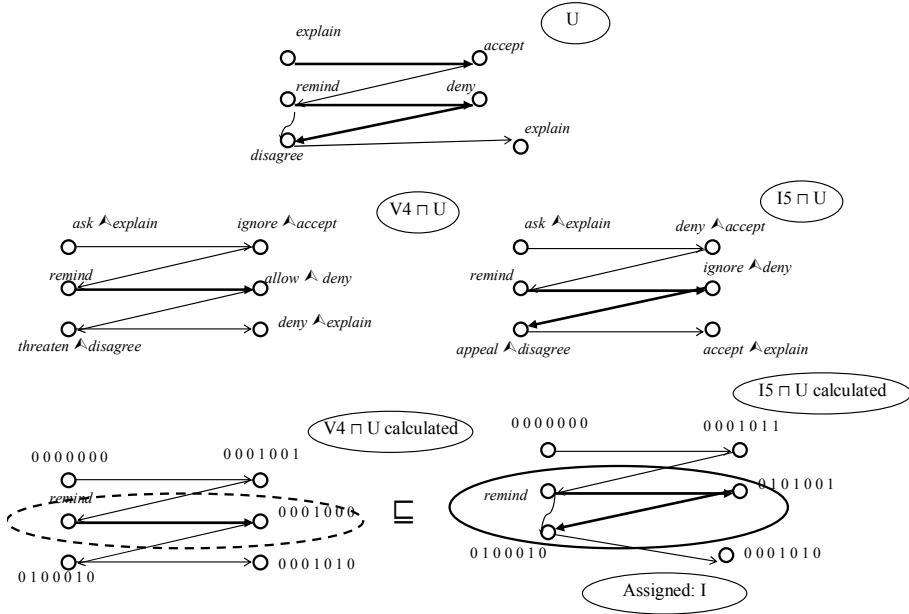


Fig. 5. A scenario with unassigned complaint status and the procedure of relating this scenario to a class

5.1 Nearest-Neighbor Classification

The following conditions hold when a scenario graph U is assigned to a class (we consider positive classification, i.e., to valid complaints, the classification to invalid complaints is made similarly):

- 1) U is similar to (has a nonempty common scenario subgraph of) a positive example R^+ .
 - 2) For any negative example R^- , if U is similar to R^- (i.e., $U \cap R^- \neq \emptyset$) then $U \cap R^- \subseteq U \cap R^+$. 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 scenario from the positive class should be higher than the similarity between U and each negative example (i.e., representative of the class of invalid complaints).
- Condition 2 implies that there is a positive example R^+ such that for no R^- one has $U \cap R^+ \subseteq R^-$, i.e., there is no counterexample to this generalization of positive examples.

Let us now proceed to the example of a particular U in Figure 5 on the top. The task is to determine whether U belongs to the class of valid complaints (on the left of

Figure 4) or to the classes of invalid complaints (on the right); these classes are mutually exclusive.

We observe that V_4 is the graph of the highest similarity with U among all graphs from the set $\{V_1, \dots, V_5\}$ and find the common subscenario $U \sqcap V_4$. Its only thick arc is derived from the thick arc between vertices with labels *remind* and *deny* of U and the thick arc between vertices with labels *remind* and *allow* of V_4 . The first vertex of this thick arc of $U \sqcap V_4$ is $remind \wedge remind = remind$, the second is $allow \wedge deny = \langle 0 0 0 1 0 0 0 \rangle$ ($U \sqcap V_4$ is calculated at the left bottom). Other arcs of $U \sqcap V_4$ are as follows: that from the vertex with the label *remind* to the vertex with the label $\langle 0 0 0 1 0 0 0 \rangle$; the arc from the vertex with the label $\langle 0 0 0 1 0 0 1 \rangle$ to the vertex with the label *remind*; the arc from the vertex with the label $\langle 0 0 0 1 0 0 0 \rangle$ to the vertex with the label $\langle 0 1 0 0 0 1 0 \rangle$. These arcs are thin, unless both respective arcs of $U \sqcap V_4$ are thick (the latter is not the case here). Naturally, common subscenario may contain multiple steps, each of them may result in the satisfaction of conditions 1) - 2) for the class assignment above.

Similarly, we build the common subscenario $U \sqcap I_5$; I_5 delivers the largest subgraph (two thick arcs) in comparison with I_1, I_2, I_3, I_4 . Moreover, $U \sqcap V_4 \sqsubseteq U \sqcap I_5$, this inclusion is highlighted by the ovals around the steps. Condition 2 is satisfied. Therefore, U is an invalid complaint as having the highest similarity to invalid complaint I_5 .

In [7, 11] we considered a learning model from [3] formulated in FCA terms. As applied to scenarios, this model is described as follows. Given similarity (meet) operation \sqcap on pairs of scenarios that defines a semilattice, sets of positive and negative examples, a (+)-hypothesis is defined as similarity of several positive examples which does not cover any negative example (for the lack of space we refer to [7, 11] for exact definitions). (-)-hypotheses are defined similarly. Now an undetermined scenario is classified positively if it contains (in terms of \sqsubseteq) a positive hypothesis and does not contain any negative hypothesis.

Finally, we discuss briefly the complexity of the approach. Generally, even testing \sqsubseteq relation is an NP-complete problem. However, our scenario graphs are usually not large, the number of vertices not exceeding 20-30. In the simplest case where the vertex labels are incomparable, the large number of vertex labels reduces practical complexity, since a vertex can be matched only to a vertex with the same label. As for computing similarity operation \sqcap , the realization of which involves several testing of \sqsubseteq , with our modest computation resources (PC with 1.7 GHz and 1GB RAM) and training sample with 40 examples, it was feasible to compute classifications in 3-4 hours using a lower part of the concept lattice of vertex labels (labels with common generalization being too general were considered dissimilar).

6 Evaluations

In this section we present the results of preliminary evaluation of our classification model. Firstly, we evaluate the accuracy of our nearest-neighbors technique given the above dataset Figure 4. For each of ten scenarios, we set its class as unknown and verify if it can be related to its class properly, building common subscenarios with

four representatives of its class and five foreign scenarios. Only scenarios I2 and V3 can be neither assigned to the proper class nor to a foreign class; the rest of scenarios were properly assigned.

Our further evaluation involved an improvement of existing software for processing customer complaints, called *ComplaintEngine*, available for download at <http://www.dcs.bbk.ac.uk/~galitsky/ComplaintEngineSuite.html>. Five attributes of mental actions, selected for the model presented in this paper, were indeed selected to improve the accuracy of scenario recognition, given the particular set of complaints from our database of formalized complaints. Our database primarily originates from the data on financial sector, obtained from the website of publicly available textual complaints PlanetFeedback.com.

Currently, *ComplaintEngine* uses anti-unification procedure to find a similarity between scenarios. Machine learning of *ComplaintEngine* uses the JSM-type plausible reasoning [3] augmented with situation calculus, reasoning about mental states and other reasoning domains. *ComplaintEngine* applies domain-independent anti-unification to formulas that include enumeration of mental actions in time.

As expected, graph representation of scenarios and employed nearest-neighbors technique allowed noticeable improvement of complaint recognition accuracy. Judging on the restricted dataset of 80 banking complaints (40 complaints make the training set and 40 complaints have to be classified), the performance of *ComplaintEngine* was improved by 6% to achieve the resultant recognition accuracy of 89%. Relating a scenario to a class, *ComplaintEngine* is capable of explaining its decision by enumeration of similar and dissimilar scenarios, as well as particular mental actions which led to its decision.

Since such accuracy was achieved by manual adjustment of the model of multi-agent scenario, we expect it to be much lower for other complaint domains. However, we believe that the role of improved machine learning technique for functioning in a new complaint domain will be substantial.

We would like to briefly familiarize the reader with the functionality of *Complaint Engine* [6]. The user interface to specify a complaint scenario is shown at Figure 6a. Figures 6b and 6c depict the fragments of this form where complainant selects his mental actions and mental actions of his opponent (a company) respectively. Mental actions are selected from the list of twenty or more, depending on the industry sector of a complaint. The parameters of mental actions are specified as text in the Interactive Form; however they are not present in the formal graph-based scenario representation.

Having performed the justification of complaint validity, *ComplaintEngine* sets the list box for complaint status at “unjustified”. *ComplaintEngine* provides the explanation of its decision, highlighting the cases which are similar to U (unjustified), and which are different from U (justified). Moreover, *ComplaintEngine* indicates the mental actions (steps) that are common for U and other unjustified complaints to further back up its decision.

A similar form to Figure 6 is used for a complainant to file a complaint, and for a company to store complaints, analyze them, determine validity, explain how the decision has been made, and finally advise on a strategy for complaint resolution (see the demo at www.dcs.bbk.ac.uk/~galitsky/CLAIMS/ComplaintEngineSuite.zip).

input your complaint

Your problem in one sentence (or choose from the list)
inadequate deposit

Your initial request
 In person (with no appointment) [v]
 date of submission []
 essence of request []
 you explain that my cheque I wrote after I made a deposit bounced []
 also, you none that []
 and you none that []

Your second request/iteration
 In person (with no ap) [v] *Check if it is a response to the issue addressed in the same line on the right above*
 date of submission []
 essence of request []
 you remind that [] unfairly charged an overdraft fee a month ago in a []
 also, you disagree that [] their fee and wanted it deposited back to my acc []
 and you none that []

Your third request/iteration
 Request an appointme [v] *Check if the fact below is the consequence of (is caused by) the fact above*
 date of submission []
 essence of request []
 you none that []
 also, you none that []
 and you none that []

Complaint status: **unsubstitid** [v]

Add Complaint Features to KB	Run complaint validity accessor	Initialise complaint validity accessor	Show steps, important for decision	Show other cases which led to decision
			Clear highlighted features	Clear highlighted cases

Fig.6a. The screen-shot of the Interactive Complaint Form where the complaint scenario U from Figure 4 is specified.

Your initial request
 way of submission In person (with no apptmen) [v]
 date of submission []
 essence of request []
 you explain that my cheque I wrote after I made a deposit bounced []
 also, you none that []
 and you none that []

Your second request/iteration
 In person (with no ap) [v] *Check if it is a response to the issue addressed in the same line on the right above*
 date of submission []
 essence of request []
 you remind that [] unfairly charged an overdraft fee a month ago in []
 also, you disagree that [] their fee and wanted it deposited back to my ac []
 and you none that []

Your third request/iteration
 Request an appointme [v] *Check if the fact below is the consequence of (is caused by) the fact above*
 date of submission []
 essence of request []
 you none that []
 also, you none that []
 and you none that []

Fig. 6b: The left pane of the interactive complaint form. Here a complainant specifies her mental actions and their parameters.

A complainant has a choice to use the above form or to input complaint as a text so that the linguistic processor processes the complaint automatically and fills the form for her. Using the form encourages complainants to enforce a logical structure on a complaint and to provide a sound argumentation. After a complaint is partially or fully specified, the user evaluates its consistency. *ComplaintEngine* indicates whether the current complaint (its mental component) is consistent or not, it may issue a warning and an advice concerning improvement of the logical structure of this complaint.

Initial response you received No response

way of response: Request an appointment by

date of response: []

essence of response: []

your tutor: accept that usually takes some time to process the depo

also, she/he: none that

and she/he: none that

Second response you received No response

way of response: Request an appointment by

date of response: []

essence of response: []

your tutor: deny that the overdraft fee was disclosed in my account

and she/he: explain that nothing can be done at this point

and she/he: none that

Third response you received No response

way of response: Request an appointment by

date of response: []

essence of response: []

your tutor: none that

also, she/he: none that

and she/he: none that

Note: A callout bubble points to the 'your tutor' row of the first form, stating: 'Check if it is a response to the issue addressed on the same line on the left'.

Fig. 6c: The right pane of the interactive complaint form. Here a complainant specifies the mental actions and their parameters of his opponent (a company).

When the complainant is satisfied with the response of *ComplaintEngine*, he submits the completed form. The other option is if a user observes that it is not possible to file a reasonable complaint, it may be dismissed at this early stage by the complainant.

7 Conclusions

In this paper we proposed a Nearest Neighbors-based approach to relate a formalized conflict scenario to the class of valid and the class of invalid complaint scenarios. The representation language is that of labeled directed acyclic graphs with generalization

operator on them. We considered the concept lattice of mental actions and showed how the procedure of relating a complaint to a class can be implemented.

This is an initial attempt to build a machine learning technique for formal scenarios as graphs. The further steps of the research along the line of machine learning for multiagent scenarios will be as follows:

- Developing a more precise representation languages for scenarios of multiagent interactions; adding more features to scenario representation in addition to temporal and causal links;
- In terms of applications, proceeding beyond the domain of customer complaints;
- Performing comparison with other classification techniques.

Building a framework for comparative analysis of formal scenarios, one or another way to express the similarity between the main entities has to be employed. In our earlier studies we approximated the meanings of mental entities using their definitions via the basis of *want-know-believe*. However, building the concept lattice for mental actions was found to be more suitable on the way to define a concept lattice for scenarios themselves.

Also, here we suggested a novel approach to building a semantic network between linguistic entities in the basis of selected attributes. The choice of attributes in this study is motivated by the task of scenario comparison; these attributes may vary from domain to domain. Twenty selected mental entities are roughly at the same level of generality – there are “horizontal” semantic relations between them.

We believe the current work is one of the first targeting machine learning in such domain as multiagent interactions described in natural language. A number of studies have shown how to enable BDI-agents with learning in a particular domain (e.g. information retrieval). In BDI settings the description of agents’ attitudes is quite limited: only their beliefs, desires and intentions are involved. Moreover, just the automated (software) agents are addressed. In this paper we significantly extended the expressiveness of representation language for agents’ attitudes, using twenty mental actions linked by a concept lattice. The suggested machinery can be applied to an arbitrary domain including inter-human conflicts, obviously characterized in natural language.

The preliminary evaluation of our model shows it is an adequate technique to handle such complex objects (both in terms of knowledge representation and reasoning) as mental actions of scenarios of multiagent interactions. Nearest Neighbors approach was found suitable to relate an inter-human conflict scenario to a class. Evaluation using our limited dataset, as well as the dataset of formalized real-world complaints showed a satisfactory performance.

Acknowledgements The second and third authors were supported by the Russian Foundation for Humanities, project no. 05-03019a.

References

1. Bratman, M.E.: Intention, plans and practical reason. Harvard University Press (1987)
2. Fagin,R., Halpern, J.Y., Moses,Y., Vardi,M.Y. : Reasoning about knowledge *MIT Press*, Cambridge, MA, London, England (1995)
3. Finn, V.K. : Plausible Reasoning in Systems of JSM-type, *Itogi Nauki I Techniki, Seriya Iformatika*, 15, 54-101, 1991 [in Russian].
4. Galitsky, B. : A Library of Behaviors: Implementing Commonsense Reasoning about Mental World. 8th Intl Conf on Knowledge-Based Intelligent Info Syst. (2004)
5. Galitsky, B.: Natural Language Question Answering System: Technique of Semantic Headers. *Advanced Knowledge International*, Adelaide, Australia (2003)
6. Galitsky, B. and Tumarkina, I.: Justification of Customer Complaints using Emotional States and Mental Actions. *FLAIRS* Miami, FL. (2004)
7. Ganter, B. and Kuznetsov, S.O.: Pattern Structures and Their Projections, Proc. 9th Int. Conf. on Conceptual Structures, ICCS'01, G. Stumme and H. Delugach, Eds., Lecture Notes in Artificial Intelligence, vol. 2120 (2001) 129-142
8. Ganter, B. and Wille R.: Formal Concept Analysis. Mathematical Foundations, Springer (1999)
9. Guerra-Hernandez, A.I, Fallah-Seghrouchni,A. E. and Soldano, H.: Learning in BDI Multi-agent Systems CLIMA IV - Computational Logic in Multi-Agent Systems Fort Lauderdale, FL, USA (2004)
10. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann (1993)
11. Kuznetsov S.O.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. In: J. Zytow, J. Rauch (eds.), Proc. Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD'99, Lecture Notes in Artificial Intelligence, vol. 1704 (1999) 384-392
12. Laza, R. and Corchado, J. M.: CBR-BDI Agents in Planning. *Symposium on Informatics and Telecommunications (SIT'02)*, Sevilla, Spain, September 25-27, pp. 181-192 (2002).
13. Mitchell, T.: Machine Learning, McGraw-Hill (1997)
14. Muller, H.J. Dieng, R. (eds.): Computational Conflicts: Conflict Modeling for Distributed Intelligent Systems. Springer-Verlag New York (2000)
15. Olivia,C., Chang, C.F., Enguix, C.F. and Ghose A.K.: Case-Based BDI Agents: an Effective Approach for Intelligent Search on the World Wide Web. *Intelligent Agents in Cyberspace. AAAI Spring Symposium* (1999)
16. Shanahan, M.: Solving the frame problem. MIT Press (1997)
17. Sowa, J.: Conceptual Graphs, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA (1984)
18. Stone, P., Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robotics*, 8(3) (2000) 345-383
19. Weiss, G., Sen, S.: Adaptation and Learning in Multiagent Systems. *Lecture Notes in Artificial Intelligence*, Vol. 1042. Springer-Verlag, Berlin Heidelberg New York (1996)
20. Wooldridge, M.: Reasoning about Rational Agents. The MIT Press Cambridge MA (2000)
21. Yevtushenko, S.A.: <http://www.sf.net/projects/conexp>. Last accessed April 7 2005.
22. Cohen, P.R, Levesque, H.J.: Performatives in a Rationally Based Speech Act Theory. *Proceedings of the 28th conference on Association for Computational Linguistics* (1990) 79-88
23. Searle, J.: Speech Acts: An Essay in the Philosophy of Language, Cambridge, Eng.- Cambridge University Press (1969)
24. Bach, K. and Harnish, R.M. Linguistic Communication and Speech Acts, Cambridge, Mass.: MIT Press (1979)
25. Searle, J.: Expression and Meaning: Studies in the Theory of Speech Acts. Cambridge University Press. (1979)