Learning Adversarial Reasoning Patterns in Customer Complaints

Boris A. Galitsky and Josep Lluis de la Rosa

University of Girona, Catalonia (Spain)

bgalistky@hotmail.com, peplluis@eia.udg.edu

Abstract

We propose a mechanism to learn communicative action structure to analyze adversarial reasoning patterns in customer complaints. An efficient way to assist customers and companies is to reuse previous experience with similar agents. A formal representation of customer complaints and a machine learning technique for handling scenarios of interaction between conflicting human agents are proposed. It is shown that analyzing the structure of communicative actions without context information is frequently sufficient to advise on complaint resolution strategies. Therefore, being domain-independent, the proposed machine learning technique is a good complement to a wide range of customer response management applications where formal treatment of inter-human interactions is required.

Introduction

Automating customer complaints processing is an important area of knowledge management. Retailers and service providers may profit from software services for customer complaints, as they allow to handle complaints faster, providing the possibility of feedback analysis and data mining capabilities on the basis of a complaint database. On the other hand, software tools for automatic complaint processing like ComplaintEngine (Galitsky 2007) allow reducing costs in complaint processing, improving communication with demanding customers and impressing the customer audience with complaint intelligence technologies. In such a setting, automated decision-support agents are important, as they can be integrated as a part of an automated infrastructure for handling complaints, interacting with customers as a part of the overall online business. Recent research (Philips-Wren 2005) has shown that being harassed by a adversarial environment, decision makers in decision support technologies for real-time and uncertain decision problems (as those related to customer complaints processing) often ignore crucial information, use inefficient strategies, and generate fewer alternatives. Examples of such problems are such as medical emergencies, traffic flow, military applications and customer relation management. These decision problems often require up-to-the-minute information, dynamic response and qualitative conflict resolution, primarily reasoning about mental states and communicative actions of involved parties.

In this paper we present the representation machinery needed for modeling conflict scenarios associated with customer complaints and propose a machine learning approach for classifying scenarios of human-agent conflicts in customer complaint situations. Scenarios will be represented as labeled directed acyclic graphs, where arcs denote the flow of interaction between two parties in a conflict. Given a scenario S, we use Nearest Neighbors (Mitchell 1997) as a technique to relate that particular scenario to the class of valid or invalid argumentation scenarios, on the basis of finding common subscenarios (subgraphs) by means of similarity matching. As we will see, this technique can be implemented in a stand-alone mode or used in combination with deductive reasoning or simulation.

Formalizing Complaint Scenarios

When modeling scenarios of inter-human conflict it is worth distinguishing communicative/physical *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), or to *moving* and *withdrawing* (physical actions). It has been shown that an adequate description of the world can be performed on the basis of communicative entities and merging all other physical action into a constant predicate for an arbitrary physical action and its resultant physical state (Galitsky 2003). In our approach we characterize a sequence $[s_1,s_2,\ldots s_k]$ of communicative states for an scenario via the set of mental actions that would unambiguously lead to these mental states. Hence we approximate an *inter-human interaction scenario* as a sequence $[a_1, \ldots,a_n]$ of communicative actions, ordered in time, with a defeat relation between some of them. Scenarios are simplified to allow for effective matching among them by means of graphs. In such graphs, communicative actions 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 a human agent. An arc (oriented edge) denotes a sequence of two actions.

In our simplified model of communication semantics communicative actions will be characterized by three parameters: (1) *agent name* (agent identifier) (2) *subject* (information transmitted, an object described, etc.), and (3)

- *(Pro) I* ***explained*** *that I made a deposit, and then wrote a cheque 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.*

Fig. 1: A conflict scenario with attack relations

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 thin arcs link vertices that correspond to communicative actions with different subjects. We will make explicit conflict situations in which the cause of one communicative action M_1 "attacks" the cause or subject of another communicative action M_2 via an *argumentation arc* (or *argumentation link*) between the vertices for these communicative actions. This attack relationship expresses that the cause of the first communicative action (starting point of the arc) defeats the subject or cause of the second communicative action (final point of the arc). 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 in Fig. 1. This text represents 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 boldface. Some expressions appear underlined, indicating that they are *defeating* earlier statements. Fig. 2 shows the associated graph, where straight thick and thin arcs represent temporal sequences, 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 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-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 as disclosed in my account information defeats the client's assertion *due to a bank error*. Similarly, the expression I made a deposit well in advance 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".*

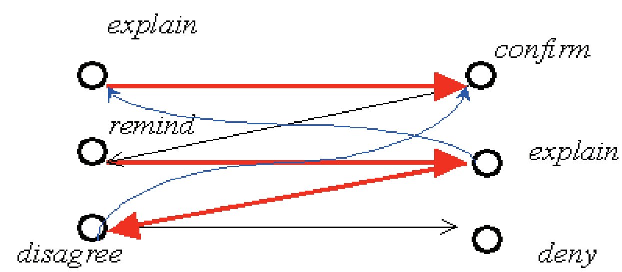

Fig. 2: Graph for a conflict scenario (Fig. 1)

It can be noticed that this complaint scenario is not sound because it seems that the complainant does not understand the procedure of *processing the deposit* nor distinguishes it from an *insufficient funds situation*. However, this scenario itself and its associated argumentation patterns do not have surface-level explicit inconsistencies, if one abstracts from the domain-specific (banking) knowledge. At the first sight, the complainant's plot in terms of communicative actions seems normal, and the complainant's arguments are not defeated. Nevertheless, by looking deeper into the case without taking into account banking knowledge a problem becomes visible: rather then accepting the opponent's *confirmation* about subject *S*, the complainant switches from *S* to another subject *S'* (*reminds* about another transaction), and *disagrees* with the opponent's explanation of this new subject, mixing both of them. Moreover, from the complainant's perspective, his opponent reacts with *denial* to his *disagreement*. In other words, the complainant disagrees with what has already been explained but at the same time "defeats" what has granted as confirmed, which is a suspicious argumentation pattern.

Our aim is to classify a new complaint scenario **without** background knowledge, on the basis of a dataset of scenarios for each class. We intend to automate the above analysis given the formal representation of the graph. Such graph was 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, as we will see later. Conflict scenarios will be formalized on the basis of user-input text, so that there can be multiple communicative actions per step (for example *I disagreed… and suggested…*). Such patterns are very common in our conflict scenarios: the former communicative action describes how an agent receives a message (*accept, agree, disagree, reject*, etc.) from another party, and the latter one describes the attitude of this agent to that communicative action, initiating a request (*suggest*, *explain*, etc.) or reacting to the other party's action. Frequently, such actions are assumed but not mentioned explicitly, and they can be deduced from the context. We will approximate each communication step using four semantic components, namely an **agent identifier** (identifies whether the communicative action is performed by the proponent or the opponent in the dialogue), a c**ommunicative action**, the **subject of communicative action** (which issue is this communicative action referring to) and the **cause for subject** (reason for referring to the subject).

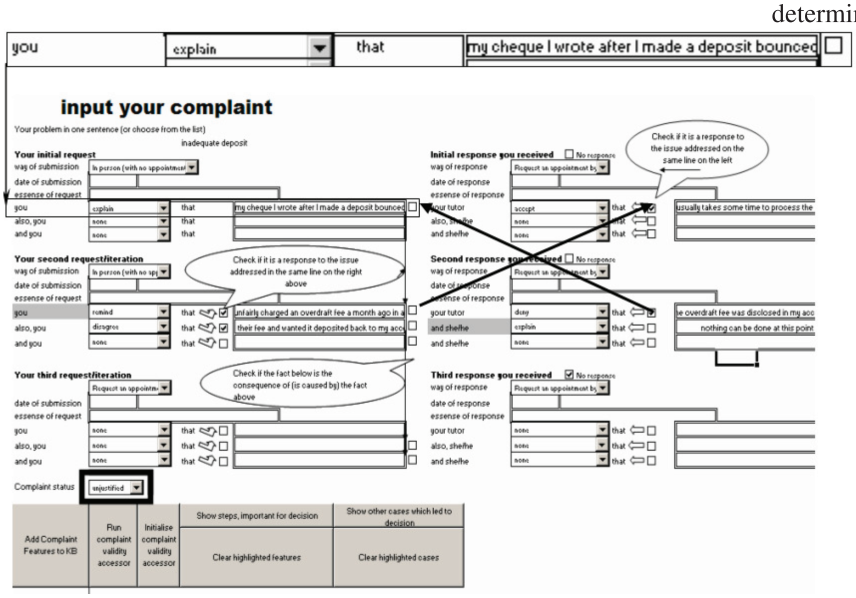

Fig. 3: Interactive Complaint Form (screenshot). A single interaction step is shown enlarged at the top

Next we will briefly outline the functionality of *Complaint Engine* (Galitsky 2007, Galitsky et al 2010), the customer complaint platform used for testing the proposed approach. The user interface to specify a *complaint scenario* is shown in Fig. 3. A complainant (e.g., a customer) selects his communicative actions (on the left) and communicative actions of his opponent (e.g., a company, on the right) respectively. Communicative actions are selected from a list of twenty or more, depending on the industry sector of the complaint. The parameters of communicative actions are specified as text in the Interactive Form (even though they are not present in the formal graph-based scenario representation). When filling in a complaint form, the user specifies implicitly a complaint scenario, modeled as a graph as discussed before. Communicative actions selected by the user in the list boxes constitute the vertices of such a graph, whereas check boxes on the right of the list boxes are used to specify whether the incoming arc is **thick** (**checked**) or thin (unchecked). Check boxes linked with a vertical line are used to specify *argumentation links* between the respective events. After performing the justification of complaint validity, *ComplaintEngine* sets the list box for complaint status at "*unjustified*" ("*justified*", resp.), indicating whether the complaint proceeds or not.

ComplaintEngine provides the explanation of this decision by highlighting the cases which are similar to the one to be classified, and which are different from it. Moreover, *ComplaintEngine* indicates the communicative actions (steps) that are common for it and other complaints to further back up its decision. *ComplaintEngine* is useful for companies as it can store complaints, analyze them, determine their validity and advise on a general strategy for complaint resolution, using the graph representation.

It must be remarked that a complainant has the choice to use the above form or to input complaint as a *text*, and a specialized linguistic tool processes that text and fills in the form for him/her. However, using the form as a "template" encourages complainants to enforce a logical structure on their complaints. Moreover, in contrast to communicative actions, it is too hard for current automated text-processing technology to reveal defeat relationships from text. In that respect the template proves to be particularly useful, as argumentation links can only be defined via the form using arrows. After a complaint is partially or fully specified, the user evaluates its consistency. *ComplaintEngine* indicates whether the current complaint is consistent or not (according to its communicative component), and it may issue a warning or advice concerning improvement of the logical structure of this complaint. When the complainant is satisfied with the response of *ComplaintEngine*, he/she can submit the completed form.

Learning semantics of communicative actions in adversarial environment

Previous implementations of *ComplaintEngine* were restricted to analysis of scenarios as sequence of communicative actions **without** a graph representation (Galitsky et al 2011). Our current approach makes the treatment of sequences of communicative actions more accurate and augments it with attack relations between the subjects of these communicative actions. Therefore, a special machine learning technique to operate with such representations was required, and a formal framework for comparison (finding similarities) between communicative actions is to be developed. Analysis of conflict scenarios is based on previous complaints which have been analyzed by experts, and characterized as *valid* or *invalid*. In order to assess the status of a new complaint, we will analyze the structure of the underlying communicative actions and argumentation patterns to check if they are similar to some previous ones assigned by an expert as valid.

Comm. 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

Fig. 4: Communicative actions in S_{freq} and associated attribute values.

It must be remarked that, in contrast to logical frameworks for defeasible argumentation (Chesnevar 2000), we do not require a mathematically formalized criterion on why one natural language expression defeats another. It is up to the user who specifies a negotiation history that one argument of himself attacks another one of his opponent. For example, the statement *"I made a deposit well in advance"* attacks the statement *"it usually takes a day to process the deposit"* in the case when access to this deposit fails, and it is up to a complainant to specify the respective attack relation in accordance to her belief.

The theory of *speech acts* (Searle, 1969, Austin 1962) is one of the most promising approach to categorizing communicative actions in terms of their roles. Following (Bach and Harnish 1979), we consider four categories of illocutionary speech acts with major representatives: *stating, requesting, promising* and *apologizing*. Each speech act is related to a single category only in the framework of the speech act theory; however for our purpose each speech act extracted from text as a lexical unit may belong to multiple categories (see Fig. 4). A number of approaches have attempted to discover and categorize how the *attitudes* and speech acts of participants in a dialogue are related to each other. Applying machine learning to the attitudes and speech acts, we are primarily concerned with how these approaches can provide a unified and robust framework to find a similarity between the speech acts in the context of understanding customer complaints. To implement such a machine learning approach we had first to identify a set S_{freq} of those communicative actions which are most frequently used for representing conflict (Fig. 5), on the basis of a structured database of previous complaints. Furthermore, to capture the similarity between communicative actions, we introduced five different attributes, each of which reflects a particular semantic parameter for communicative activity:

- *Positive/ negative attitude* expresses whether a communicative 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 communicative 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 communicative 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 communicative state so that a particular communicative action is chosen. Thus we have high knowledge/confidence(1), lack of

knowledge/confidence(-1), neither or both are possible (0).

- *Intense / relaxed mode* tells about the potential emotional load, high emotional load (1), low (-1), neutral or either is possible (0).

These attributes were on their turn associated with the different communicative actions in S_{freq} as shown in Fig. 4. Note that out of the set of meanings for each communicative action (entity), we merge its subset into a single meaning, taking into account its relations to the meanings of other communicative actions. To represent the hierarchy of communicative actions by a *concept lattice* (Ganter & Wille 1999), we scale nominally the first, second, and fourth attributes. The third and fifth attributes are already two-valued. Thus, the scaled context has eight attributes, resulting in a concept lattice. Some selected nodes are provided with descriptions of the corresponding "intents" and "extents" subscribed to show how certain communicative actions are semantically related to each other. The concept lattice illustrates the semantics of communicative actions, and shows how the choice of natural language semantics for communicative entities covers the totality of meanings in the knowledge domain of interaction between agents in a complaint scenario. Figure 5 displays the similarities between communicative actions expressed via their attributes. The graph presents the communicative actions and their similarities. If the values of respective attributes of communicative actions are the same, it will remain in the similarity result, and otherwise the similarity is denoted by "*x*": $1 \cup 1 = 1$, $0 \cup 0 = 0$, $0 \cup 1 = x$. Only close similarities are shown: deviation by one attribute (solid box) and by two attributes (dashed box). As an exception, the similarity

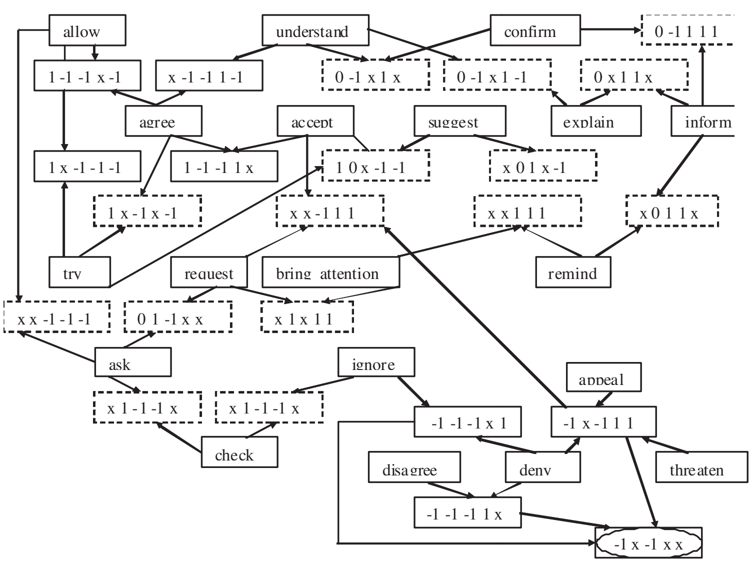

Fig. 5: Clustering communicative actions by similarities

edge of *confirm-inform* is shown on the top left (*x x* 1 *x* -1). In particular we can distinguish two main "clusters": a) a cluster of communicative actions associated with negative attitudes which do not supply information (on the right bottom). Here the communicative actions are similar to each other: they deviate from *deny* by one attribute out of five. Also, the difference between *deny*, *appeal* and *threaten* is the second attribute only, and therefore their similarity is expressed by the same vertex (-1 *x* -1 1 1). Moreover, three similarity vertices for this cluster converge to the similarity vertex for the whole cluster (-1 *x* -1 *x x*), highlighted by an ellipse; b) a cluster for the rest of communicative actions which are connected with each other and linked with the above cluster by the *deny/accept* link.

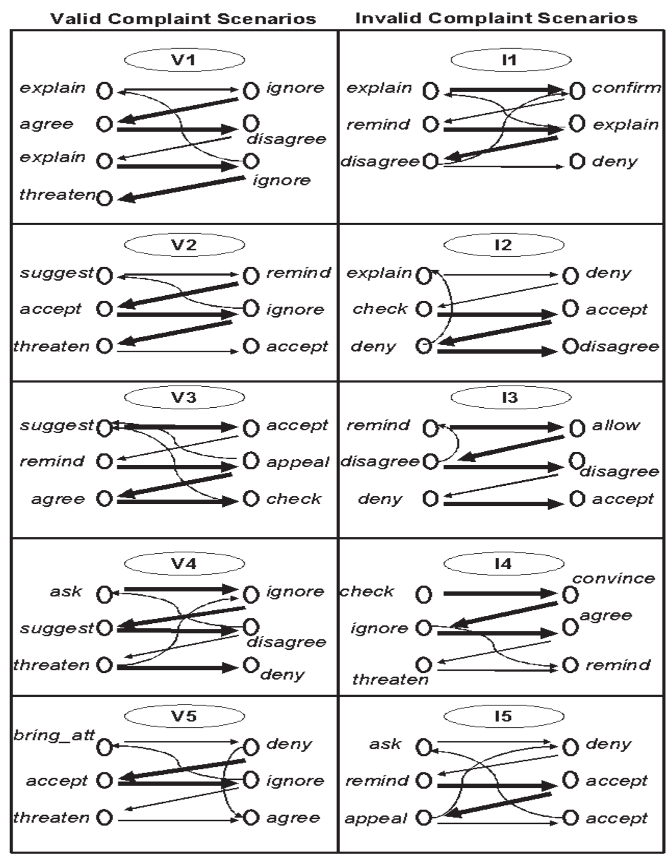

Fig. 6: The training set of scenarios

As stated before, our approach is based on a *scenario dataset* which constitutes a set of training examples for classifying new complaint scenarios. In this dataset complaint scenarios are classified into two categories: those scenarios which show a good attitude of the complainant (consistent plot with proper argumentation, a *valid complaint*) and those which show a bad attitude of a complainant (inconsistent plot with certain flaws, implausible or irrational scenarios, an *invalid complaint*). These two kinds of scenarios are depicted in Fig. 6. Each training scenario includes between two and six interaction steps, each consisting of communicative 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 communicative actions with the same subject are linked with **thick** arcs. Thus, for example, *suggest* from scenario V_2 (Fig. 6) is linked by a thin arc to the communicative action *remind*, whose argument is not logically linked to the argument of *suggest* (the subject of suggestion). The first step of V_2 includes *remind-accept-ignore-threaten*; these communicative actions have the same subject (it is not specified in the graph of conflict scenario). The vertices of these communicative actions with the same subject are linked by the thick arcs. We can summarize the constraints for a scenario graph as detailed below:

(1) All vertices are fully ordered by the temporal sequence (earlier-later);
(2) Each vertex is either assigned with the proponent (drawn on the right side of each graph in Fig. 6) 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 *defeat* 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, first we consider formal definitions of labeled graphs and domination relation on them (for space reasons we do not provide all details here; see e.g. Ganter, S. Kuznetsov 2003). A *generalization* Z of a pair of scenario graphs X and Y (or the *similarity* between X and Y), 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: first, to be matched, two vertices from graphs X and Y must denote communicative actions of the same agent; second, each common subgraph from Z must contain at least one thick arc.

Now we are ready to introduce the algorithm of how to relate a scenario to a class, given the examples from positive and negative classes (Fig. 6). The following conditions should hold for assigning a scenario graph to the class of "valid complaints" (we consider classification to the positive class, i.e., valid complaints, as the classification to invalid complaints is made analogously): (1) U is similar to (has a nonempty common scenario subgraph of) a positive example $R+$, and (2) For any negative example R^-, if U is similar to R^- (i.e., $U*R^- \neq \emptyset$) then $U*R^- \subseteq U*R^+$. This last 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). Note that condition 2 implies that there is a positive example R^+ such that for no R^- one has $U*R^+ \subseteq R^-$, i.e., there is no counterexample to this generalization of positive examples.

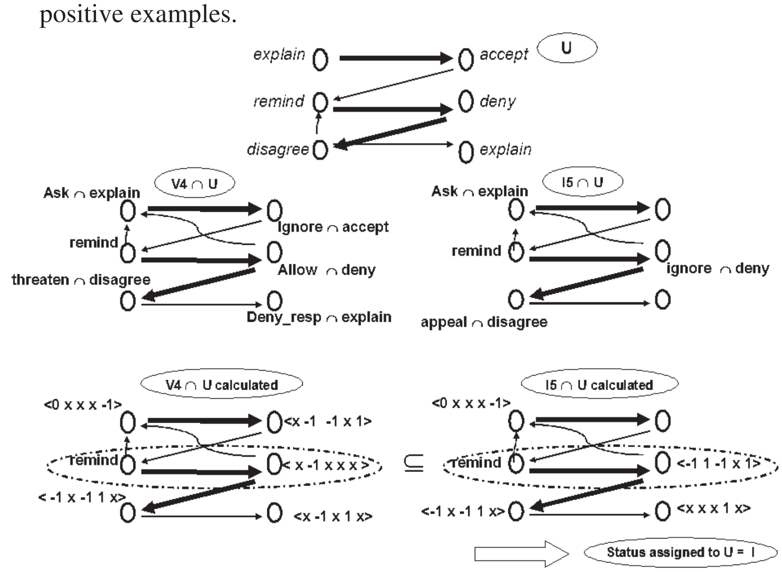

Fig. 7: A scenario with unassigned complaint status and the procedure of relating this scenario to a class

Let us now proceed to the example of a particular U as shown in Fig.7 on the top. The task is to determine whether U belongs to the class of valid complaints (Fig. 7, left) or to the class of invalid complaints (Fig. 7, right); clearly, 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 $\{V_1, \ldots V_5\}$ and find the common subscenario $U*V_4$. 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 $U*V_4$ is *remind ^ remind=remind*, the second is *allow ^ deny = <x -1 x x x >* ($U * V_4$ is calculated at the left bottom). Naturally, common sub-scenario may contain multiple steps, each of them may result in the satisfaction of conditions (1) and (2) for the class assignment above. Similarly, we build the common subscenario $U*I_5$; I_5 delivers the largest subgraph (two thick arcs) in comparison with I_1, I_2, I_3, I_4. Moreover, we have that $U*V_4 \subseteq U*I_5$ (this inclusion is highlighted by the ovals around the steps), so that Condition 2 is satisfied. Therefore, U is an invalid complaint as having the highest similarity to invalid complaint I_5.

Evaluation of complaint classification

We performed the comparative analysis of relating scenarios to a class taking into account (a) communicative actions only; (b) argument structure only and (c) both communicative actions and argument structure. Such an analysis sheds a light on the possibility to recognize a scenario without background knowledge, which is a typical situation in real-world complaint analysis. Comparing the contributions of communicative actions and argumentation to relating a scenario to a class, we explored the high-level roles of these components and the peculiarities of their inter-connection. Revealed rules for typical valid and typical invalid complaints in a given domain would help to reveal corresponding behavior patterns, which are essential for complaint handling personnel.

We conducted a two-level evaluation of communicative actions and argumentation patterns. On the first level, we considered a limited dataset of formalized real-world complaint scenarios and performed their classification. For each complaint scenario, we set its class to unknown and verified if it can be related to its class properly, building common subscenarios with the representatives of its class and foreign scenarios. On the second level, we evaluated the functionality of the currently available complaint processing system, augmented with argumentation analysis. Processing an extended set of complaints, we compare a recognition accuracy of the base system, (which takes into account communicative actions only) with that of the augmented system.

Our reduced dataset included 58 complaints which we selected as typical and sufficiently complex to be represented as a graph with at least six vertices (42 valid and 16 invalid complaints, 50% of each is a training set and 50% have to be classified). We obtained the following recognition accuracy for this dataset: 64% with communicative actions only, 43% with argumentation only, and 78% by combining communicative actions and argumentation. Hence argumentation improves the classification accuracy for this dataset by about 22%, and the stand-alone argumentation analysis delivers less than 50% classification accuracy. We believe that the relation between the above percentages is an important outcome compared with these percentages as such being domain-dependent. It is also worth mentioning that these recognition settings assume relating a scenario to a class and providing a background for the decisions.

Our second-level evaluation for our complaint database primarily originates from the data on financial sector, obtained from the website of publicly available textual complaints PlanetFeedback.com. For the subset of this dataset which includes the complaints with argumentation, the performance of *ComplaintEngine* was improved by 16% to achieve the resultant recognition accuracy of 91%. However, for the overall dataset the improvement was 7% only. Nevertheless one may conclude that taking into account argumentation is important for accurate assessment of complaint validity.

Related Work and Conclusions

In this paper we have proposed a Nearest Neighbors-based approach to improve automated processing of customer complaints in the *ComplaintEngine* software platform. We have shown how communicative actions along with argumentation patterns can be successfully modelled in terms of graphs, capturing similarities among them to assess their validity. In earlier studies (Galitsky et al 2007, Galitsky et al 2011) we approximated the meanings of communicative entities using their definitions via the basis of "want-know-believe". However, building the concept lattice for communicative actions was found to be more suitable, particularly as a way to define a concept lattice for scenarios themselves.

To the best of our knowledge, there is no similar approach in targeting machine learning techniques for such domain as assessing the validity of customer complaints. A number of studies have shown how to enable multiagent systems with learning in a particular domain (e.g. information retrieval) and how to enable them with argumentation capabilities. In particular, machine learning frameworks for operating with rich conflict scenarios (as those involving inter-human interactions) have not been yet explored, although a number of case-based reasoning approaches have been suggested to treat the scenarios of interaction in the belief-desire-intention (BDI) agent model (Rao & Georgeff 19950, as shown in (Olivia et al 1999, Stone & Veloso 2000). However, in such approaches the description of agents' attitudes is reduced to their beliefs, desires and intentions, without involving a richer language for communicative entities as proposed in our approach. In this paper we significantly extended the expressiveness of representation language for attitudes, using different communicative 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.

Adversarial reasoning has been defined (Kott & McEneaney 2006) as a series of computational approaches to inferring and anticipating perceptions, intents and actions of opponents. The authors argue that adversarial reasoning goes beyond the boundaries of game theory and must also include such areas as cognitive modeling, control theory, AI planning and others. Authors also describe a battle planning system that focuses on brigade-level ground operations and involves adversarial reasoning.

According to (Pelta & Yager 2009), a possible action against an adversary is to make decisions that are intended to confuse him, although proponent's rewards can be diminished. It is assumed that making decisions in an uncertain environment is a hard task. However, this situation is of upmost interest
in the case of adversarial reasoning, since it is beneficial to confuse the adversary in situations of repeated conflicting encounters. Using simulations, the use of dynamic vs. static decision strategies were analyzed and it turned out that the presence of an adversary may produce a decrease of, at least, 45% with respect to the theoretical best payoff.

As a final conclusion we can say that the preliminary evaluation of our model of adversarial argumentation attached to subjects of formalized communicative actions shows that it is an adequate technique to handle such complex objects as communicative actions of scenarios for multiagent interactions (both in terms of knowledge representation and reasoning). Evaluation experiments using our limited dataset, as well as the dataset of formalized real-world complaints showed a satisfactory performance, although the most promising results seem still to be ahead.

References

Galitsky, B., Galitsky, B., Kovalerchuk, B. Kuznetsov SO, Learning common outcomes of communicative actions represented by labeled graphs. Intl Conference on Concept Structures Sheffield UK July 22-27 LNCS 4604 pp 387-400.

G. Philips-Wren, L. J. (eds), Intelligent Decision Support Systems in Agent-Mediated Environments, IOS Press, 2005.

T. Mitchell, Machine Learning, McGraw-Hill, 1997.

B. Galitsky, Natural Language Question Answering System: Technique of Semantic Headers, Advanced Knowledge International, 2003.

Galitsky, B. and Josep Lluis de la Rosa. Concept-based learning of human behavior for customer relationship management. Special Issue on Information Engineering Applications Based on Lattices. Information Sciences. Volume 181, Issue 10, 15 May 2011, pp 2016-2035 (2011).

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

J. Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, Cambridge, UK, 1969.

J. L. Austin, How To Do Things with Words, Oxford University Press, Oxford, UK, 1962.

K. Bach, R. Harnish, Linguistic Communication and Speech Acts, Cambridge (MA): MIT Press, 1979.

B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer Verlag, 1999.

B. Ganter, S. Kuznetsov, Hypotheses and version spaces, in: Proc. 10th Int. Conf. on Conceptual Structures. In A. de Moor, W. Lex, and B.Ganter, Eds., LNAI vol. 2746, pp. 83-95 (2003).

A. Rao, M. Georgeff, BDI agents: From theory to practice, Tech. rep. 56, Australian Artificial Intelligence Institute (1995).

C. Olivia, C. Chang, C. Enguix, A. Ghose, Case-based BDI agents: an effective approach for intelligent search on the world wide web intelligent agents in cyberspace, in: Proc. of the AAAI Spring Symposium, 1999.

P. Stone, M. Veloso, Multiagent systems: A survey from a machine learning perspective, Autonomous Robotics 8 (3) (2000) 345–383.

Alexander Kott, William M. McEneaney. Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind. Chapman & Hall/CRC. 2006.

David A. Pelta and Ronald R. Yager. Dynamic vs. static decision strategies in adversarial reasoning. IFSA-EUSFLAT 2009.