An Argumentation based Approach to Multi-Agent Learning

Santiago Ontañón
Cognitive Computing Lab (CCL)
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, USA
santi@cc.gatech.edu

Enric Plaza
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
enric@iiia.csic.es

Abstract

This paper addresses the issue of learning from communication among agents that work in the same domain, are capable of learning from examples, and communicate using an argumentative framework. We will present (1) an argumentation framework for Case-Based Reasoning agents and (2) an individual policy for agents to generate arguments and counterarguments (including counterexamples). We focus on argumentation between two agents, presenting an interaction protocol (AMAL2) that allows agents to learn from counterexamples and a preference relation to determine the joint outcome when individual predictions are in contradiction. The experimental evaluation shows that argumentation-based joint predictions and learning examples from communication both improve over individual predictions.

Introduction

The more challenging issues in applying Machine Learning (ML) techniques to Multiagent Systems (MAS) are those that are distinctive of MAS, namely communications, coordination, and competition. Although ML has been applied both to improve coordination in MAS and competition among agents, less research has been devoted to learn from communication among agents. In this paper, we address two issues: (1) using communication to establish a collaboration mechanism among Case-Based Reasoning agents in order to improve their performance, and (2) learning from communication to provide CBR agents with a new information source allowing them to improve their performance.

In this paper we consider a scenario with two agents that (1) work in the same domain using a shared ontology, (2) are capable of learning from examples, and (3) communicate using an argumentative framework. A CBR agent can predict the solution of new problems using knowledge learnt from past experience. Additionally, in our scenario an agent may obtain information by arguing with some other agent about the correct prediction for a problem. For this purpose, we will present a two fold approach consisting of (1) an argumentation framework for CBR agents, and (2) an individual policy to generate arguments and counterarguments.

Existing argumentation frameworks for multiagent systems are based on deductive logic. An argument is seen as a logical statement, while a counterargument is an argument offered in opposition to another argument (Chesñevar & Simari 2000; S. Parsons 1998); agents use a preference relation to resolve conflicting arguments. However, logic-based argumentation frameworks assume agents with predetermined knowledge and preference relation. In this paper, we focus on argumentation framework where both knowledge and preference relation are learned from experience.

Having learning capabilities allows agents to effectively use a new form of counterargument, namely the use of *counterexamples*. Counterexamples offer the possibility of agents learning *during* the argumentation process. Moreover, CBR agents allow the design of techniques that use learnt experience to generate adequate arguments and counterarguments. Specifically, we will need to address two issues: (1) how to define a preference relation over two conflicting arguments, and (2) how to define a technique to generate arguments and counterarguments.

This paper presents a case-based approach to address both issues. The agents use case-based reasoning to learn from past cases in order to predict the outcome of a new situation. We propose an argumentation protocol, AMAL2, that supports two agents in reaching a joint prediction over a specific situation or problem — moreover, the reasoning needed to support the argumentation process will also be based on cases. In particular, we present two *case-based measures*, one for determining preference relation among arguments and another for establishing the policy for generating arguments and counterarguments. Finally, we evaluate (1) if argumentation between 2 CBR agents can produce a joint prediction that improves over individual learning performance and (2) if learning from the counterexamples conveyed during the argumentation process increases the individual performance with just those cases being interchanged among the CBR agents.

In the remainder of this paper we are going to introduce the multi-agent CBR ($\mathcal{M}$AC) framework and the notions of justifications and justified predictions. After that, we will provide a specific definition of arguments and counterarguments that we will use in the rest of the paper. Then, we will define a preference relation between contradicting arguments. After that, specific policies to generate both argu-

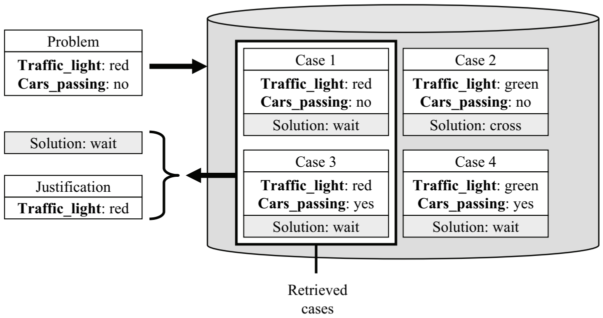

Figure 1: An example of justification generation.

ments and counterarguments will be presented. Using the previous definitions, the AMAL2 interaction protocol for argumentation is introduced afterwards. Finally, we will show an empirical evaluation of our approach.

Justifications in Multi-Agent Learning

In this section we are going to define the multi-agent learning framework in which our research is performed. A *Multi-Agent Case Based Reasoning System* ($\mathcal{M}$AC) $\mathcal{M} = \{(A_1, C_1), ..., (A_n, C_n)\}$ is a multi-agent system composed of $\mathcal{A} = \{A_i, ..., A_n\}$, a set of CBR agents, where each agent $A_i \in \mathcal{A}$ possesses an individual case base C_i. Each individual agent A_i in a $\mathcal{M}$AC is completely autonomous and has access to its individual and private case base $C_i = \{c_1, ..., c_m\}$. Agents in a $\mathcal{M}$AC system are able to individually solve problems, but they can also collaborate with other agents to solve problems.

In this framework, we will focus analytical tasks, (e.g. classification), where solving a problem means selecting a class from an enumerated set of solution classes. In the following we will note the set of all solution classes by $\mathcal{S} = \{S_1, ..., S_K\}$. Therefore, a *case* $c = \langle P, S\rangle$ is a tuple containing a case description P and a solution class $S \in \mathcal{S}$. In the following, we will use the terms *problem* and *case description* indistinctly, and we will use the dot notation to refer to elements inside a tuple. e.g., $c.S$ refers to the solution class of a case c.

A *justification* built by a CBR method after determining that the solution of a particular problem P was S_k is a description that contains the relevant information from the problem P that the CBR method has considered to predict S_k as the solution of P. In particular, CBR methods work by retrieving similar cases to the problem at hand, and then reusing their solutions for the current problem, expecting that since the problem and the cases are similar, the solutions will also be similar. Thus, if a CBR method has retrieved a set of cases $C_1, ..., C_n$ to solve a particular problem P the justification built will contain the relevant information from the problem P that made the CBR system retrieve that particular set of cases, i.e. it will contain the relevant information that P and $C_1, ..., C_n$ have in common.

For example, Figure 1 shows a justification build by a CBR system for a toy problem (in the following sections we will show justifications for real problems). In the figure, a problem has two attributes (*Traffic_light*, and *Cars_passing*), the retrieval mechanism of the CBR system notices that by considering only the attribute *Traffic_light*, it can retrieve two cases that predict the same solution: *wait*. Thus, since only this attribute has been used, it is the only one appearing in the justification. The values of the rest of attributes are irrelevant, since whatever their value the solution class would have been the same.

In general, the meaning of a justification is that all (or most of) the cases in the case base of an agent that satisfy the justification (i.e. all the cases that are *subsumed* by the justification) belong to the predicted solution class. In the rest of the paper, we will use $\sqsubseteq$ to denote the subsumption relation. In our work, we use LID (Armengol & Plaza 2001), a CBR method capable of building symbolic justifications such as the one exemplified in Figure 1. The idea of LID is to start with an empty description that subsumes all the cases of the case base, and keep refining it until it only subsumes the problem at hand and a reduced set of cases in the case base from which a solution can be predicted. The refined description generated by LID is exactly the justification. When an agent provides a justification for a prediction, the agent generates a *justified prediction* $J = \langle A, P, S, D\rangle$ where agent A considers S the correct solution for problem P, and that prediction is justified a symbolic description D such that $J.D \sqsubseteq J.P$. Justifications can have many uses for CBR systems (Ontañón & Plaza 2003; Plaza, Armengol, & Ontañón 2005). In this paper, we are going to use justifications as arguments.

Argumentation for Multi-agent Learning

An *argument* α generated by an agent A is composed of a statement S and some evidence D supporting S as correct. In the remainder of this section we will see how this general definition of argument can be instantiated in specific kind of arguments that the agents can generate. In the context of $\mathcal{M}$AC systems, agents argue about the correct solution of new problems and can provide two kinds of information: a) specific cases $\langle P, S\rangle$, and b) justified predictions: $\langle A, P, S, D\rangle$. In other words, CBR agents can provide as arguments either specific cases or generalizations induced from that cases. Using this information, and having in mind that agents will only argue about the correct solution of a given problem, we can define three types of arguments: justified predictions, counterarguments, and counterexamples.

A *justified prediction* α is generated by an agent A_i to argue that A_i believes that the correct solution for a given problem P is $\alpha.S$, and the evidence provided is the justification $\alpha.D$. In the example depicted in Figure 1, an agent A_i may generate the argument $\alpha = \langle A_i, P, \mathsf{Wait}, (\mathsf{Traffic\_light} = \mathsf{red})\rangle$, meaning that the agent A_i believes that the correct solution for P is Wait because the attribute Traffic_light equals red.

A *counterargument* β is an argument offered in opposition to another argument α. In our framework, a counterargument consists of a justified prediction $\langle A_j, P, S', D'\rangle$ generated by an agent A_j with the intention to rebut an argument α generated by another agent A_i, that endorses a solution class different from that of α for the problem at

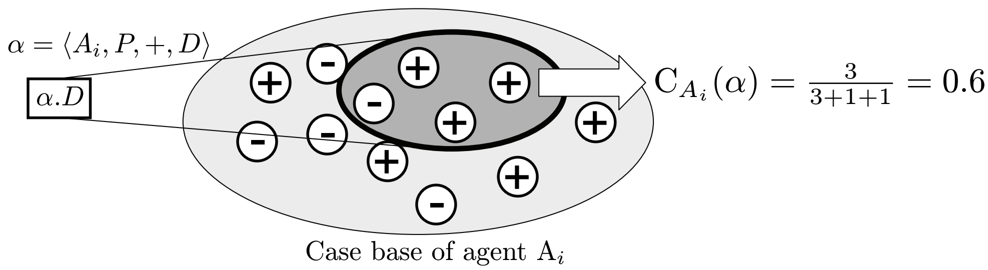

Figure 2: Confidence of arguments is evaluated by contrasting them against the case bases of the agents.

hand and justifies this with a justification D'. In the example depicted in Figure 1, if an agent generates the argument $\alpha = \langle A_i, P, \mathsf{Walk}, (\mathsf{Cars\_passing = no})\rangle$, an agent that thinks that the correct solution is Stop might answer with the counterargument $\beta = \langle A_j, P, \mathsf{Stop}, (\mathsf{Cars\_passing} = \mathsf{no} \wedge \mathsf{Traffic\_light = red})\rangle$, meaning that while it is true that there are no cars passing, the traffic light is red, and thus the street cannot be crossed.

A *counterexample* c for an argument α is a case that contradicts α. Thus a counterexample is a kind counterargument stating that a specific argument α is false, and the evidence provided is the case c. Specifically, for a case c to be a counterexample for α, the following conditions have to be met: $\alpha.D \sqsubseteq c$ and $\alpha.S \neq c.S$.

By exchanging arguments and counterarguments (including counterexamples), agents can argue about the correct solution of a given problem. For this purpose, they need a specific interaction protocol, a preference relation between contradicting arguments, and a decision policy to generate counterarguments (and counterexamples).

Preference Relation

The argument that an agent provides might not be consistent with the information known to other agents (or even to some of the information known by the agent that has generated the justification due to noise in training data). For that reason, we are going to define a case-based preference relation over contradicting justified predictions based on assessing a *confidence* measure for each justified prediction.

The confidence of justified predictions is assessed by the agents via an process of *examination of justifications* (Ontañón & Plaza 2003). The idea behind examination is to count how many of the cases in an individual case base *endorse* that justified prediction, and how many of them are *counterexamples*. The more endorsing cases, the higher the confidence; and the more the counterexamples, the lower the confidence.

To examine a justified prediction α, an agent obtains the set of cases contained in its individual case base that are subsumed by $\alpha.D$. After an agent A_i has examined a justified prediction α, he obtains the *aye* and *nay* values:

- $Y_\alpha^{A_i} = |\{c \in C_i | \; \alpha.D \sqsubseteq c.P \wedge \alpha.S = c.S\}|$ is the number of cases in the agent's case base *subsumed* by the justification $\alpha.D$ that belong to the solution class $\alpha.S$,
- $N_\alpha^{A_i} = |\{c \in C_i | \; \alpha.D \sqsubseteq c.P \wedge \alpha.S \neq c.S\}|$ is the number of cases in the agent's case base *subsumed* by justification $\alpha.D$ that do *not* belong to that solution class.

When two agents A_1 and A_2 want to assess the confidence on a justified prediction α made by one of them, each of them examines the argument and sends the *aye* and *nay* values obtained to the other agent. Then, both agents can assess the joint confidence for the justified prediction α:

$$\mathsf{C}(\alpha) = \frac{Y_\alpha^{A_1} + Y_\alpha^{A_2} + 1}{Y_\alpha^{A_1} + Y_\alpha^{A_2} + N_\alpha^{A_1} + N_\alpha^{A_2} + 2}$$

i.e. the confidence on a justified prediction is the number of endorsing cases divided by the number of endorsing cases plus counterexamples found by each of the two agents. Notice that we add 1 to the numerator and 2 to the denominator, this is the Laplace correction to estimate probabilities, that prevents arguments to have excessive confidence when assessing confidence with a small number of cases. Figure 2 illustrates the individual evaluation of the confidence of an argument, in particular, three endorsing cases and one counterexample are found in the case base of agents A_i, giving an estimated confidence of 0.6

Thus, the preference relation used in our framework is the following one: a justified prediction α is preferred over another one β if $C(\alpha) \geq C(\beta)$.

Moreover, notice that an agent might estimate the confidence of an argument by only using its own case base. For instance, an agent A_i might estimate the confidence on a prediction as: $\mathsf{C}_{A_i}(\alpha) = \frac{Y_\alpha^{A_i}}{Y_\alpha^{A_i} + N_\alpha^{A_i} + 1}$. Notice that $\mathsf{C}_{A_i}(\alpha)$ is an estimation of $\mathsf{C}(\alpha)$, that might be used by individual agents to individually compare arguments.

Generation of Arguments and Counterarguments

In our framework, arguments are generated using CBR. Any CBR method able to provide a justified prediction can be used to generate arguments. For instance, LID (Armengol & Plaza 2001) is a suitable learning method.

Thus, when an agent wants to generate an argument endorsing that a specific solution class is the correct solution for a given problem P, it generates a justified prediction as explained previously.

When an agent A_i generates a counterargument β to rebut an argument α, A_i expects that β is preferred over α. With that purpose, in this section we are going to present a specific policy to generate counterarguments based on the *specificity* criterion (Poole 1985). The specificity criterion is widely used in deductive frameworks for argumentation, and states that between two conflicting arguments, the most specific should be preferred since it is, in principle, more informed. Thus, counterarguments generated based on the specificity criterion are expected to be preferable (since they are more informed) to the arguments they try to rebut. However, there is no guarantee that such counterarguments will always win, since we use a preference relation based on confidence.

Therefore, when an agent wants to generate a counterargument β to an argument α, it will generate a counterargument that it is more specific than α.

The generation of counterarguments using the specificity criterion imposes some restrictions over the learning method, although LID can be easily adapted for this task. For instance, LID is an algorithm that generates a description starting by the empty term and heuristically adding features to that term. Thus, at every step, the description is more specific, and the number of cases that are subsumed by that description is reduced. When the description only covers cases of a single solution class, LID terminates and predicts that solution class. To generate a counterargument to an argument α LID just has to use as starting point the description $\alpha.D$ and proceed to specialize it further. In this way, the justification provided by LID will always be subsumed by $\alpha.D$, and thus the resulting counterargument will be more specific than α. However, notice that LID may sometimes not be able to generate counterarguments, since LID may not be able to specialize the description $\alpha.D$ any further, or because the agent A_i has no case in C_i that is subsumed by $\alpha.D$.

Moreover, agents may also try to rebut the other agent's arguments using counterexamples. Specifically, when an agent A_i wants to rebut an argument α, the following policy is used: (1) The agent A_i tries to generate a counterargument β more specific than α (in our experiments agents use LID). If the A_i succeeds, β is sent to the other agent as a counterargument of α. If not, (2) then A_i searches for a counterexample $c \in C_i$ of α in its individual case base C_i. If a case c is found, then c is sent to the other agent as a counterexample of α. If no counterexamples are found, then A_i cannot rebut the argument α.

Argumentation-based Multi-Agent Learning

In this section we will present the Argumentation-based Multiagent Learning protocol for 2 agents (AMAL2). The idea behind AMAL2 is to allow a pair of agents to argue about the correct solution of a problem, arriving at a join solution that is based on their past learning and the information they exchange during argumentation.

At the beginning of the protocol, both agents will make their predictions for the problem at hand. Then, the protocol establishes the rules allowing one of the agents in disagreement with the prediction of the other to provide a counterargument. Later, the other agent can respond with another counterargument, and so on.

The AMAL2 protocol among two agents A_1 and A_2 to solve a problem P works in a series of rounds. We will use t to denote the current round (initially $t = 0$) and $H_t = \langle \alpha_1^t, \alpha_2^t \rangle$ to denote the pair of predictions that each agent holds at a round t. Initially, each agent makes its individual prediction. Then, the confidence of each prediction is assessed, and the prediction with the highest confidence is considered the winner. However, if the agent that has provided the prediction with lower confidence doesn't agree, it has the opportunity to provide a counterargument. Agents keep exchanging arguments and counterarguments until they reach an agreement or until no agent is able to generate more counterarguments. At the end of the argumentation, if the agents have not reached an agreement, the prediction with the highest confidence is considered the joint prediction.

The protocol starts when one of the two agents receives a problem P to be solved. That agent sends P to the other agent requesting it to argue about the correct solution of the problem. Thus, after both agents know the problem P to solve, round $t = 0$ of the protocol starts:

1. Initially, each the agent individually solves P, and builds a justified prediction (A_1 builds α_1^0, and A_2 builds α_2^0). Then, each agent A_i sends the performative $assert(\alpha_i^0)$ to the other agent. Thus, both agents know $H_0 = \langle \alpha_1^0, \alpha_2^0 \rangle$.
2. At each round t, the agents check whether their arguments in H_t agree. If they do the protocol moves to step 5, otherwise the agents estimate the overall confidence for both arguments in H_0 and use the preference relation (explained previously) to determine which argument in H_t is preferred. After that, the agent that has provided the non preferred argument may try to rebut the other agent's argument. Each individual agent uses its own policy to rebut arguments:
   - If an agent A_i generates a counterargument α_i^{t+1}, then, it locally compares α_i^{t+1} with its previous argument α_i^t by locally assessing their confidence. If $\mathsf{C}_{A_i}(\alpha_i^{t+1}) > \mathsf{C}_{A_i}(\alpha_i^t)$, then A_i considers that α_i^{t+1} is stronger than its previous argument, and A_i has to change its argument to the stronger one; therefore A_i sends in a single message the following performatives to A_j: $rebut(\alpha_i^{t+1}, \alpha_j^t)$, $withdraw(\alpha_i^t)$, $assert(\alpha_i^{t+1})$. Otherwise, since $\mathsf{C}_{A_i}(\alpha_i^{t+1}) \leq \mathsf{C}_{A_i}(\alpha_i^t)$), A_i will send only the performative: $rebut(\alpha_i^{t+1}, \alpha_j^t)$. The protocol moves to state 3.
   - If an agent A_i selects c as a counterexample of the other agent's justified prediction, then A_i sends the following performative to A_j: $rebut(c, \alpha_j^t)$. The protocol moves to step 4.
   - If no agent provides any argument the protocol moves to step 5.
3. The agent A_j that has received the counterargument α_i^{t+1}, locally compares it against its own argument, α_j^t, by locally assessing their confidence. If $\mathsf{C}_{A_j}(\alpha_i^{t+1}) > \mathsf{C}_{A_j}(\alpha_j^t)$, then A_j will accept the counterargument as stronger than its own argument, and it will send in a single message the following performatives to A_i: $withdraw(\alpha_j^t)$, $assert(\alpha_i^{t+1})$. Otherwise, since

Table 1: Accuracy of individual prediction (IP) and joint prediction (JP).

	SOYBEAN		ZOOLOGY	
	IP	JP	IP	JP
25%	48.53%	60.2%	66.34%	77.82%
50%	62.22%	75.44%	81.98%	90.49%
75%	73.62%	82.93%	86.37%	91.68%
100%	78.24%	86.19%	90.89%	91.29%

$\mathsf{C}_{A_j}(\alpha_i^{t+1}, \leq \mathsf{C}_{A_j}(\alpha_j^t))$, A_j will not accept the counterargument, and will send a reject message to A_i. The protocol moves back to state 2 in a new round $t+1$.

4. The agent A_j that has received the counterexample c generates a new argument α_j^{t+1} that takes into account c. To inform A_i of the new argument, A_j sends A_i the following performatives: $withdraw(\alpha_j^t)$, $assert(\alpha_j^{t+1})$. The protocol moves back to state 2 in a new round $t+1$.
5. The protocol ends yielding a joint prediction, as follows: if both arguments in H_t agree then their prediction is the joint prediction, otherwise the prediction in H_t with the higher confidence is considered the joint prediction.

To avoid infinite iterations, if an agent sends twice the same argument, the protocol also terminates.

Experimental Evaluation

In this section we empirically evaluate the AMAL2 argumentation protocol. We have made experiments in two different data sets from the UCI machine learning repository: *soybean* and *zoology*. The soybean data set has 307 examples and 19 classes, while the zoology data set has 101 cases and 7 classes. In an experimental run, training cases are distributed among the agents without replication, i.e. there is no case shared by two agents. In the testing stage problems arrive randomly to one of the agents.

The experiments are designed to test two hypotheses: (H1) that argumentation is a useful framework to establish cooperation among CBR agents; and (H2) that learning from communication in argumentation processes improves the individual performance of a CBR agent. Moreover, we expect that the improvement is correlated with the sample of data accessible to a CBR agent; specifically, an agent with less cases can benefit more both from argumentation-based cooperation (H1) and learning from communication (H2). Therefore, our experiments focus on how the amount of data accessible to an agent influences the degree of improvement.

Concerning H1, we ran 4 experiments where part of the data is distributed among 2 CBR agents. Table 1 shows the results of experimental runs for 25%, 50%, 75% and 100% of the training set distributed among 2 CBR agents. The results shown are the average of five 10-fold cross validation runs. For this H1 scenario learning from communication is disabled in order to evaluate the performance of the case-based argumentation process for joint prediction. The results in Table 1 show that joint prediction (JP) using AMAL2 outperforms individual accuracy (IP) of a CBR agent. Moreover, as expected, the degree of improvement is larger when the individual agents have a smaller sample of the data. For instance, in the Soybean data set with 25% of the data JP accuracy increases 11.67 over IP; while with 100% JP accuracy increases 6.97 over IP. The difference in accuracy improvement (which is smaller in Zoology) is explained by the fact that Zoology is an *easier* data set — in the sense that with a smaller percent of the data higher accuracy values can be achieved. Thus, learning tasks in more "difficult" data sets also have more to gain from cooperation using a case-based argumentation process.

Concerning H2, we ran 4 experiments similarly as before, except that now during the training phase the agents learn from the counterexamples interchanged in the process of argumentation. That is to say, when an agent A_i receives a new training case c an argumentation process is engaged in order to reach a joint prediction; those counterexamples that are exchanged in order to predict the case's solution are retained by receiving agent (i.e. both A_i and A_j can learn from communication). Case c is retained by A_i after the argumentation is finished. Table 2 shows the results of experimental runs for 25%, 50%, 75%, and 100% of the training set distributed among 2 learning agents. The results in Table 2 shows that the individual accuracy of agents that also learn from communication (ALC) outperforms that of agents learning only from their individual experience (AIL). Notice that, since we are evaluating H2, Table 2 plots individual accuracy only, and not joint prediction accuracy as Table 1 does. For instance, we can see how with 100% of the data, accuracy increases from 78.24% to 81.95% in the soybean data set, and from 90.89% to 93.47% in the zoology data set. These results show that learning from communication (in addition to learning from individual experience) improves the individual accuracy, and therefore the argumentation process provides a second source of information that can be fruitfully exploited by individual learners. Moreover, the number of cases learnt from communication (CLC) is reasonably small compared to the number of individually retained cases (CIL). For instance, notice that in with 100% of the data, only an average of 2.92 extra cases were learnt from communication. Moreover, the increment in individual accuracy shows that the argumentation process provides a good selection mechanism for determining those cases that can be useful for a particular agent.

Related Work

Concerning CBR in a multiagent setting, the first research was on "negotiated case retrieval" (Prassad, Lesser, & Lander 1995) among groups of agents. Our work on multiagent case-based learning started in 1999 (Martín, Plaza, & Arcos 1999); later Mc Ginty and Smyth (McGinty & Smyth 2001) presented a multiagent collaborative CBR approach (CCBR) for planning. Finally, another interesting approach is *multi-case-base reasoning* (MCBR) (Leake & Sooriamurthi 2002), that deals with distributed systems where there are several case bases available for the same task and addresses the problems of cross-case base adaptation. The main difference is that our $\mathcal{M}$AC approach is a way to dis-

Table 2: Accuracy with individual learning (AIL) and together with learning from communication (ALC); cases retained (CIL) and learned from communication (CLC).

	SOYBEAN				ZOOLOGY			
	AIL	ALC	CIL	CLC	AIL	ALC	CIL	CLC
25%	48.53%	51.66%	33.90	4.15	66.34%	69.11%	11.00	0.97
50%	62.22%	68.6%	67.80	8.73	81.98%	84.95%	22.00	1.80
75%	73.62%	76.78%	101.70	11.12	86.37%	88.32%	33.00	2.27
100%	78.24%	81.95%	138.15	12.93	90.89%	93.47%	45.45	2.92

tribute the *Reuse* process of CBR (using a voting system) while *Retrieve* is performed individually by each agent; the other multiagent CBR approaches, however, focus on distributing the *Retrieve* process.

Research on MAS argumentation focus on several issues like a) logics, protocols and languages that support argumentation, b) argument selection and c) argument interpretation. Approaches for logic and languages that support argumentation include defeasible logic (Chesñevar & Simari 2000) and BDI models (S. Parsons 1998). Although argument selection is a key aspect of automated argumentation (see (S. Kraus & Evenchik 1998) and (S. Parsons 1998)), most research has been focused on preference relations among arguments. In our framework we have addressed both argument selection and preference using a case-based approach.

Conclusions and Future Work

In this paper we have presented an argumentation-based framework for multiagent learning. Specifically, we have presented a protocol called AMAL2 that allows two CBR agents to argue about the solution of a given problem and we have shown how the learning capabilities can be used to generate arguments and counterarguments. The experimental evaluation shows that the increased amount of information that the agents use to solve problems thanks to the argumentation process increases their predictive accuracy, and specially when the individual agents have access to a limited amount of information. Clearly, an agent that knows all does not need external help (nor, by the way, needs to continue learning if there is no room for improvement).

The main contributions of this work are: a) an argumentation framework for CBR agents; b) a case-based preference relation over arguments, based on a confidence estimation of arguments; c) a policy to generate counterarguments and counterexamples; and d) an argumentation-based approach for learning from communication.

Moreover, the work presented in this paper concerns only pairs of agents: as future work we plan to generalize the AMAL2 protocol to work with a larger number of agents. We would like to elucidate two concerns: a) in which situations argumentation-based joint prediction can significantly increase performance (number of agents, bias in individual data, etc); and b) how learning from communication in an argumentation-based process can improve the individual performance in addition to the individual learning. In the experiments presented here a CBR agent would retain all counterexamples submitted by the other agent; however, this is a very simple *case retention policy*, and we will like to experiment with more informed policies — with the goal that individual CBR agents could significantly improve using only a small set of cases proposed by other agents.

Acknowledgments. This research was partially supported by the MID-CBR project TIC2006-15140-C03-01.

References

Armengol, E., and Plaza, E. 2001. Lazy induction of descriptions for relational case-based learning. In *ECML'2001*, volume 2167 of *Lecture Notes in Artificial Intelligence*, 13–24.

Chesñevar, C. I., and Simari, G. R. 2000. Formalizing Defeasible Argumentation using Labelled Deductive Systems. *Journal of Computer Science & Technology* 1(4):18–33.

Leake, D., and Sooriamurthi, R. 2002. Automatically selecting strategies for multi-case-base reasoning. In *ECCBR'2002*, volume 2416 of *Lecture Notes in Artificial Intelligence*, 204–219. Berlin: Springer Verlag.

Martín, F. J.; Plaza, E.; and Arcos, J.-L. 1999. Knowledge and experience reuse through communications among competent (peer) agents. *International Journal of Software Engineering and Knowledge Engineering* 9(3):319–341.

McGinty, L., and Smyth, B. 2001. Collaborative case-based reasoning: Applications in personalized route planning. In Watson, I., and Yang, Q., eds., *ICCBR*, number 2080 in LNAI, 362–376. Springer-Verlag.

Ontañón, S., and Plaza, E. 2003. Justification-based multiagent learning. In *Proc. ICML'2003*, 576–583. Morgan Kaufmann.

Plaza, E.; Armengol, E.; and Ontañón, S. 2005. The explanatory power of symbolic similarity in case-based reasoning. *Artificial Intelligence Review* 24(2):145–161.

Poole, D. 1985. On the comparison of theories: Preferring the most specific explanation. In *IJCAI-85*, 144–147.

Prassad, M. V. N.; Lesser, V. R.; and Lander, S. 1995. Retrieval and reasoning in distributed case bases. Technical report, UMass Computer Science Department.

S. Kraus, K. S., and Evenchik, A. 1998. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence Journal* 104:1–69.

S. Parsons, C. Sierra, N. R. J. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8:261–292.