A Cooperative Multiagent Architecture for Turkish Sign Tutors *

İlker Yıldırım

Department of Computer Engineering
Boğaziçi University
Bebek, 34342, Istanbul, Turkey
ilker.yildirim@boun.edu.tr

1 Introduction

A sign is a combination of hand movements, facial expressions, and head movements. A sign with only hand movements is called manual signs, whereas signs which also include head movements or facial expressions are called non-manual signs. The aim of the *Sign Language Tutoring Tool* is to help users learn isolated signs by watching recorded videos and enable them try the same signs [1]. The system records a user's video, while she is performing a sign. After analyses of user's sign performance, the system gives the user feedback both verbal and animated. The system can recognize not only manual signs, but also non-manual, complex signs, that involve both hand movements and head movements and face expressions. The system uses a classifier for recognition, by which it can assess some similarity value with a level of certainty for user's sign performance. The Turkish Sign Language Tutoring Tool is specialized for Turkish Sign Language.

The current system is a stand-alone application. However, in reality the same application will need to be run in different locations with different data. This obviously calls for a distributed architecture. A possible architecture would be to have a collection of stand-alone applications in which there is no interaction between agents. In such a system although agents may improve their classification capabilities due to learning and experience, they will not be able pass their experiences to others and they will be not be able benefit from experiences of others.

Another possible architecture would be a centralized one, in which agents are geographically seperated, but they share a common database and a common classifier. But in reality we know that agents may not be online all the time, hence they cannot access to the database and the classifier. In addition to that, videos may belong to a certain individual who do no want to share it.

Taking all these into account, we propose a cooperative multiagent system of Sign Language Tutoring Tools, which consist of many agents distributed geographically [2]. Each agent represents an agent that runs a sign language tutor.

* The author is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under grant 107E021.

Agents are connected via Internet. Each agent is associated a local database and a classifier. Agents can improve its classification performance due to its own experience. An agent may decide to include a practice sign in its training data or a sign language teacher may add a new training data. Moreover, agents can help each other classify signs by exchanging classification requests. Since agents have their own local databases they can make a decision even they are offline.

2 Possible Challenges with the Architecture

In this section we state several example cases in order to better show different challenges. We will use Figure 1 as an example setup. In this figure we have four agents. Each agent can talk with each other. Agent 1 (the agent at station 1) is asked to make a decision on user's sign performing video, i.e. the sign of "school". That is Agent 1 will either decide that the user-performed sign is "school" or not. In order to come up with a better decision Agent 1 passes the user's video of sign school to other agents and requests them to state their ideas about the user's video. The following examples depict interesting situations that require systematic decision making.

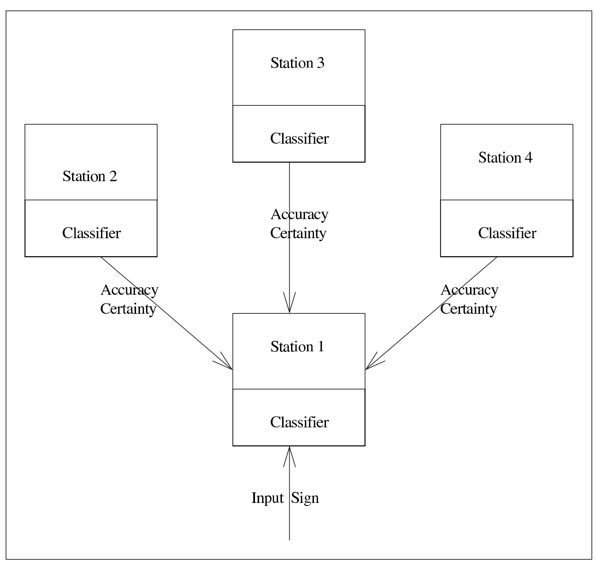

Fig. 1. A Setup for the Distributed Sign Language Recognition System

Example 1 Each agent can decide if a sign is performed *right* or *wrong*. However agents may not be confident on their decisions, hence they can request others to provide opinions (i.e. in form of votes). In Figure 1 Agent 1 receives votes for "right" from Agents 2 and 4 which means they find it similar enough to actual sign of school, whereas it receives a "wrong" vote from Agent 3 which means it does not find user's performance similar enough. In addition to ideas of other agents Agent 1 also has its vote for "right".

Example 2 The previous case assumes all agents equally knowledgeable. Actually, there are two important criteria for decision: *Accuracy:* How similar is the performance of the user to the real sign, *Certainty:* How certain about that accuracy. In Figure 1, Agent 2 says the user's video is the sign of school with a high accuracy but with a low certainty. Agent 3's response is, similar to Agent 2's, "The performed sign is very similar to sign of school but I am not sure". Apart from those Agent 4 says "I am sure that the user's performance is not the sign of school". In addition to the ideas of other agents, Agent 1 has no idea on the user's video, meaning its certainty level is very low or zero (But it has many ideas to run a weighted majority voting on it).

Example 3 Before sending user's video, Agent 1 sends the word whose sign is performed to each agent. After each agent receives the word, they have vote to elect the one from whom to request a decision for the user's performance. Hence, after all agents agree on the agent which will make the decision, Agent 1 sends the user's sign performance to that agent.

Example 4 It is very likely that we have high number of agents. In such a case, it is possible that most agents have low certainty. Hence the aggregation of their votes may influence the system in a wrong way. But instead, each agent can model others based on expertise and direct queries accordingly. Let there be several other agents in Figure 1 and let Agent 1 be requesting votes of other agents on the sign of "theater". Since agent 1 thinks that an agent knows very well signs about art and literature, it directs query accordingly, and not request from every other agent in the system.

3 Parameters of Decision Making

There are several parameters that affect an agent while it evaluates an idea of accuracy with some level of certainty about a user video:

- How well my classifier is trained for this particular sign?
- Who entered my training data, a deaf person, a sign language teacher or a sign language student?
- When my training data was entered? For instance the resolution of my training videos may be low, because they are old.
- What is the amount of data used to train my classifier?

In Example 1 each agent has an idea about whether the user's video is the sign of school, meaning right or the user's video is not the sign of school, meaning wrong. Consider we apply a plurality voting protocol which means the candidate with highest number of votes wins [3]. Our candidates are *the user's sign* *performance video is right* or *the user's sign performance video is wrong*. Each agent votes by responding to Agent 1. In Example 1 *the user's sign performance video is right* receives 3 votes whereas *the user's sign performance video is wrong* receives only 1 vote. Hence Agent 1 concludes that the user performed the sign of school right.

In Example 2 each agent responds how similar the sign of school and the user's video are similar (its accuracy), and how certain it is about that accuracy (its certainty). In Example 2 although two of agents argue that the user's video has high accuracy, Agent 1 will decide that the user's video is wrong, because Agent 4 is certain that the user's video is not the sign of school. In order to be able come up with such a conclusion, Agent 1 considers accuracy values of other agents using their certainty values.

Before considering Examples 3 and 4, an agent is required to have a model of other agents in order to enhance its decision making process. One way to make agents construct models of other agents is to give feedback about its decisions up to now. For instance Agent 1 decided that the subject performs the corresponding sign of the word "school" correctly relying on the certainty of Agent 3. After sometime, a teacher manually checks decisions of Agent 1 and recognizes that while the subject could not perform the sign of the word "school" correctly, somehow, Agent 1 accepted it. The teacher inputs that feedback to the agent, and immediately Agent 1 changes its beliefs about Agent 3 in some sense negative direction. The reverse can also happen. Models can have several dimensions, i.e. expertise on an ontological subject, how much to trust to certainty value, how much to trust to accuracy.

In Example 3, agents elect one of them to make the decision. It is equal to saying we have number of agents many candidates and number of agents many voters. We can apply any voting mechanism to elect the decision maker. Let's consider *Single Transferable Vote (STV)* [3]. In STV the winner is determined after rounds. In each round each candidate receives points as the number of votes in which it ranks highest. In each round the candidate with the lowest score is eliminated. The last remaining candidate is the winner. The name comes from that the votes are transferred from most popular to the next most popular candidate in terms of voters of the eliminated candidate. After electing the agent to make decision, Agent 1 can get this decision and reply to the subject.

In Example 4 each agent models other agents in the system. An agent, when it is requested by the user to recognize a performance, requests a subset of other agents for help. The agent decides which subset of agents to ask using its models. Since agents improve via experience, these models need to be dynamic. So we need methods for updating models of others. Since there are a lot of agents, each agent may not be able to model each other. If an agent does not know whom to ask exactly, it can ask to another agent that she thinks that may know. The queried agent may either reply an answer or *refer* to another agent.

In Figure 2, our generic multiagent protocol, an agent can request classification of a sign performance. And an agent may respond with as *correct or wrong*, *certainty and accuracy tuple*, or a *referral*.

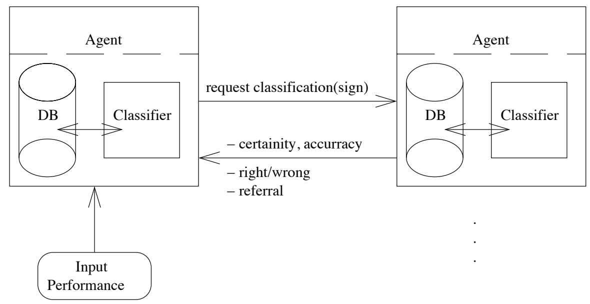

Fig. 2. Generic Multiagent Protocol

4 Discussion

In this study, we develop a cooperative multiagent system for sign language tutoring. We study different techniques for distributed decision making. After implementation of the system, we also have plans to experimentally evaluate the performance of such a system.

A training data will be distributed to each agent. Each agents training data is not disjoint from others, that means there may be intersections. After receiving its portion, each agent will train its classifier using its portion of training data. The training data may include errors or variations of the same sign on purpose, since we suppose to have such cases in reality. To test majority and weighted majority voting, we will run test data over the system and calculate the percentage of success, how many times agents made the correct decision. Whereas for model-based approaches we will measure how successful the models of agents are.

There are also some issues on which we are not clear about. The first is the compositional decision case. Agent is requested to recognize a sentence in sign language. We can assume that the sentence is divided into its parts. Agent decides on each part individually with support of other agents using one of the models described above. And then it makes a final composed decision. The question is *How to combine individual decisions?*

The second issue is dialects of the sign languages. There are dialects of Turkish Sign Language. Assume there are two different signs for the word father. And two different agents are educated for the different fathers. Then here is the problem: For the performance of the sign father one agent says right, whereas other says wrong and actually they are both saying the correct thing. How to recognize and deal with dialects?

Durfee discusses challenges in Distributed Problem Solving and Planning, using several problems and possible and proposed solutions to those problems [4]. He states that distributed problem solving requires marhsalling of distributed capabilities in the right ways which thus requires to solve the problem of how to solve together. To come up with such a solution, agents need to communicate for task sharing and result sharing. For task sharing, he discusses contract-net protocol in terms of several problems and possible solutions to those problems. He states that in order to achieve a confident and complete distributed problem solving agents should share their results.

In [5], a proposal of cooperative decision making and support system is given for general organization. There are different realizations of a cooperative decision support system: It may be a communication system between decision makers, who are people or it may be a system of decision making units in cooperation with other elements of the system. They argue that in organizations, partners should be able model others in enough depth to be able to request complex tasks from others. Further, partners should be able to go into necessary level of communication in order to use these complex detailed.

References

1. Aran, O., Ari, I., Benoit, A., Carrillo, A.H., Fanard, F.X., Campr, P., Akarun, L., Caplier, A., Rombaut, M., Sankur, B.: Signtutor: An interactive system for sign language tutoring, IEEE Multimedia, to appear (2008)
2. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review **10**(2) (1995) 115–152
3. Conitzer, V., Sandholm, T.: Vote elicitation: Complexity and strategy-proofness. In: In Proceedings of the National Conference on Articial Intel ligence (AAAI), Edmonton, Canada (2002)
4. Durfee, E.: Distributed problem solving and planning. Multi-Agent Systems and Applications (2001) 118–149
5. Soubie, J.L., Zarat, P.: Distributed decision making: A proposal of support through cooperative systems. Group Decision and Negotiation **14**(2) (2005) 147–158