COAST: Context-aware pervasive speech recognition system

Meghdad Aynehband[1] , Amir Masoud Rahmani[2], Saeed Setayeshi[3]
[1]Islamic Azad University, Dezful branch, Dezful, Iran [2]Islamic Azad University, Science and research branch Tehran, Iran, [3]Amirkabir University of Technology, Tehran, Iran
[1]aynehband@gmail,com, [2]rahmani@sr.iau.ac.ir, [3]Setayesh@aut.ac.ir

Abstract— **Context-aware applications adapt their behavior to the user current situation. This paper presents a new architecture named COAST (Context-aware Speech to text translator). Reducing user interaction and selecting the best classifier based on contexts are the primary objectives in COAST and user's privacy rules can be applied too. The contexts are categorized in two sets: system-contexts and classification contexts. The system contexts adapt systems behaviors. The Classification contexts guide COAST to select current classifiers and modify some of them. COAST can work without server to enable autonomic behavior. Clients can connect to peers to achieve more advantages such as: fault-tolerance feature, with severs connection, achieving more contexts from the other clients' resources.**

Keywords- speech recognition; context-aware; pervasive; classification;

1. INTRODUCTION

Context-aware applications adapt their behavior to the user's current situation. To obtain the user's current situation without asking him, different sensors can be used. For some applications, the sensor data can be utilized directly (e. g., the user's current location), for others, the sensor data has to be combined with other data sources—such as user profiles or map data—to derive the situation (e. g., "in a meeting"). For this, they need context information about real-world entities like objects, persons, or places [1]. The current studies usually focus on the usage of speech for interaction between users and systems [2] [3] because the speech and gesture recognition are likely to reduce interaction between users and systems. Briefly, the current speech recognition systems work in two ways: 1- only with one classifier, 2- with multi classifiers. This paper will be describing a new architecture in the second part. You will see an overview of the COAST architecture, in the second section of this paper; the third section describes the client-side architecture of the COAST and the contexts hierarchy. The server-side architecture is mentioned in the fourth section.

2. AN OVERVIEW OF THE COAST ARCHITECTURE

In this work, a new architecture is created based on context-aware pervasive computing to enable sensing current situation to improve selecting the best classifier. This method is named COAST (Context-Aware Speech to Text Translator).

COAST searches into the classifiers and, depending on current context, selects a classifier and translates speech to text by the selected classifier. Context-aware applications use context information to adapt their behavior and adjust to changing computing environments or changing user circumstances. The provisioning of context information needs to be reliable. The context information must be available when it is needed by applications. It is a challenging requirement as the sources of context information (e.g., sensors) can't succeed or may become disconnected due to communication failures or mobility of the context-aware application. A common way to provide this reliability is through redundancy of context sources. Having redundant and always active sources of context information [5], it is possible to withstand failures of some context sources; it is also possible to support mobile applications as their context queries/ subscriptions can be served by the active context information sources in the environments. This approach, however, can be very wasteful in terms of the resource usage and it will not scale to large numbers of the sensors [6]. COAST will use sensors from other clients, if they allow it for solving the problem. Another problem in the COAST is the connection between clients & server for getting classifiers. COAST can be used without server access. COAST uses a few standard classifiers and can modify some classifiers to reduce access to the server and also can change some classifiers to improve speech recognition ability in some contexts for solving these problems. Therefore COAST can be used in *ambient systems* [7] and *artificial life* [8] field. Removing user interaction and selecting the best classifier based on context are the primary aims in COAST, but user privacy rules can be applied. This is an embedded system that contains clients and a server. The clients can connect together to share sensors data and resources, and connect to server to achieve classifiers and methods for modifying the classifiers. COAST overview is shown below:

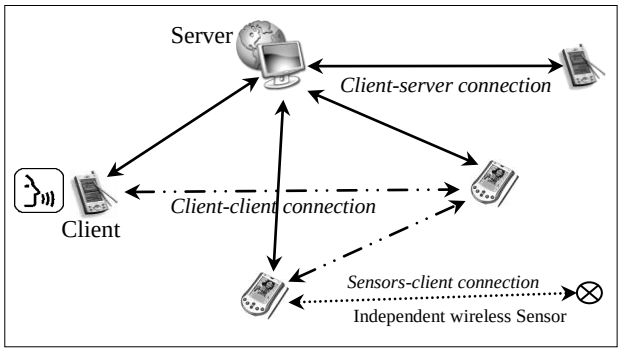

Figure 1. COAST overview

3. CLIENT-SIDE ARCHITECTURE

Clients are a 5-layer architecture, which is shown in Figure 2. In continue, the independent wireless sensors architecture is similar to clients without *Learning Layer*, *classifier creator layer* and *classifier selector layer*, all tiers are described separately as follow:

3- 1) Sensor layer

Information is retrieved to achieve these purposes in this layer: Necessary data to create contexts, Accent detection, Emotion recognition (time, date, noise, weather …), Speaker / listener recognition (health, age, sex …), positioning sensors (GPS …) and all of the things that help the COAST to create contexts. The context hierarchy will be described in the next section.

3- 2) Context creator Layer

The context creator layer must create contexts to be used by other layers. Some available sensors convert their sensed situation to a specific format and pass it to the next layer, in this step.

SYSTEM CONTEXTS

There are some contexts in this section that can be efficient to improve performance in *resource sharing*, *privacy and policy fields*.

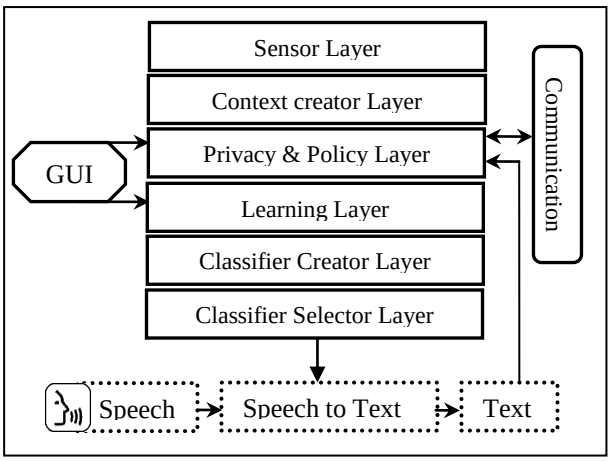

Figure 2. Client-side architecture of the COAST

CLASSIFICATION CONTEXTS

The contexts are used to create classifiers in this section, and this has two separate parts. First part is *Classifier Modifier* that modifies the classifiers and makes new ones; the second part is *classifier getter* that gets the suitable classifier based on current contexts.

Classifier Modifier

The contexts are created in this section; allow us to modify some classifier. This modification approximately improves the speech recognition ability such as Emotion recognition.

Classifier getter

In this section, contexts are created to select and achieve available classifiers from current system or the other systems. For instance, COAST is going to convert a speech to text, at the first step, COAST discovers a person (from his/her voice or face or RFID or etc.), and downloads the specific classifier from the server, but these aren't very simple works. Speaker recognition contexts contain person recognition, accent detection, race detection, location detector. ***The Listener recognition***

The people who speak to others with different languages depend on the listeners' personality, race and the other parameters.

Emotion Recognition

This context helps COAST to achieve the classifiers depending on the Emotion. For instance, COAST can get the rainy classifier for the rainy weather (or noisily classifier for noisily environment and etc.).

- ***COASTT contexts***
  - ***System contexts***
    - Sharing resources
    - Privacy
    - Policy
  - ***Classification contexts***
  - **Classifier Modifier**
    - *Emotion recognition*
  - **Classifier getter**
    - *Speaker recognition*
    - Person recognition
    - Accent detection
    - Race detection
    - *Location detector*
    - Area-Block Classifier
    - City Classifier
    - State Classifier
    - Country Classifier
    - Region Classifier
  - *Listener recognition*
  - *Emotion Recognition*

Figure 3. Client contexts hierarchy

3- 3) Privacy & Policy Layer

This Layer includes two sub layers, privacy sub layer and policy sub layer. These sub layers issues rules to access the resources such as contexts, communication access, and data of the sensors. Policy sub layer controls the resources which don't disclose user's state. Privacy sub Layer controls the specific contexts that are related to the user's privacy such as: some PH sensors (like as blood pressure sensors, user's authentication data, user's location and etc.).

3- 4) Learning Layer

This section contains five separate parts including creator/modifier of basic rules, creator/modifier of classifiers, creator/modifier of privacy and policy's rules, creator/modifier of energy rules and creator/modifier of fault tolerance rules.

3- 5) Classifer creator Layer

This layer creates classifiers and passes them to the next level to select one of them. Creating the classifiers includes two methods: first- getting classifier without modification by *Classifier getter contexts*, second- modifying classifiers based on *Classifier Modifier contexts*. There are some methods to get classifiers and described as follow:

- The Classifiers are available in the client system.
- The classifiers are accessible in available client systems.
- The classifiers are stored in the server.
- The above classifiers may be modified or new classifiers can be made.

The Created classifiers are passed to the next level to select one of them.

3- 6) Classifier selector layer

The *classifier selector* blocks are shown in Figure 4.

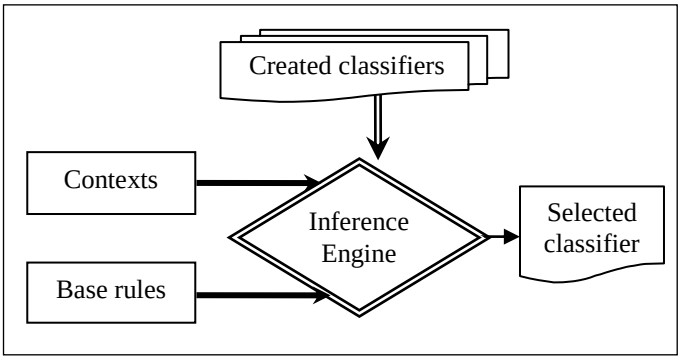

Figure 4. Classifiers selector blocks

a) Contexts

The sensor's retrieval data must be transformed into a specific format and sent to the other layers for usage. Contexts can be achieved from other clients too.

b) Base rules

These rules can be static or dynamic. Static rules can be created once, at system design and can be modified by GUI and dynamic rules can be modified inelegancy depends on contexts. An important factor is the contexts accessibility. COAST must ignore the effects of contexts that aren't accessible.

c) Created classifier

The classifiers (that are created in the *classifier creator layer*) are used here.

d) Inference engine

This engine is also a classifier, which selects a classifier among all created classifiers, based on *contexts* and *base rules* to translate speech to text by selected classifier. *Heuristic Algorithms*, *Fuzzy controllers*, *Neuron-Fuzzy systems*, *Neural network*, *Genetic Algorithm* and in the general *Artificial Intelligent* can be employed for this matter.

4. SERVER-SIDE ARCHITECTURE

COAST has server-side architecture that can improve quality and reduce client's load (such as storage, creating new classifiers that depend on new situations and etc.). The clients can work without servers too. In that situation a client must achieve classifiers itself or from other clients and must store achieved classifies in its storage. The server-side architecture is shown in figure 5 and this section of paper describes the server-side architecture's layers:

4- 1) Communication Layer

Connecting to the server can be assumed by different ways. One way can be sensor network model. Server can be as a *sink* and the clients can be as sensors. Another communication way is *internet* and *intranet*. It's important that the wireless communication is an inseparable part of COAST (client-side particularly). Clients can connect to server via other clients too.

4- 2) Policy Layer

This layer controls client access to server parts. Two concerns exist in the policy topics. These concerns are stated as follow:

a) Access to the classifers

The access permissions to the classifiers are verified in this section. The accounting system can be implied by this layer and authentication is enabled now.

b) Create a new or modify the currents classifiers

Can the client change the current classifiers or create a new classifier? After categorizing the classifiers, this question can be answered.

PERSONAL CLASSIFIERS

These classifiers belong to a client and the client can modify its classifiers. It's not an important concern.

LOCAL CLASSIFIERS

It's very sensitive for COAST to modifying the local classifiers because all local resident clients' performance will be changed. For example assume that some systems are going to achieve the Persian standard classifier, and the Turkish accent is connected to the server in a local place. If this manner is extended to other places and modified The Standard Persian Classifiers, terrible consequences will be appeared. Therefore this task is devolved to *Learning Layer*.

STANDARD CLASSIFERS

If other classifiers can't be accessible, the standard classifiers for languages will be used. These classifiers are very sensitive and must be modified very carefully.

4- 3) Learning layer

This layer must modify classifiers and contexts (such as base rules). These classifications are:

Classifiers:

Creating personal classifies:

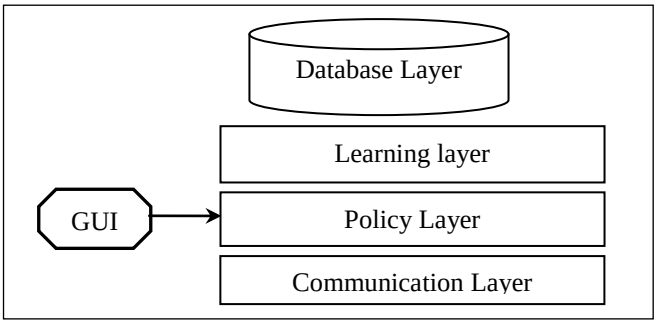

Figure 5. Server-side Architecture

One basic classifier is assumed at user's classifiers and saved in the server. This classifier can be used when the user is recognized by authentication. For instance, a user enters a system and logs into (by person recognition context) it, and then his/her personal classifier is used for speech recognition purposes.

Modifying personal classifiers

The users' accent will be changed permanently in his/her life duration. It is separate from the context's modification. The personal classifiers shall be modified if these events occurred.

Creating the classifiers based on contexts

The personal classifiers based on contexts are stored. For instance, if a voice of a person had been changed by catching a cold, then a new classifier will be created and stored in the server to use again later.

Classifier modification parameters based on contexts

This section must store the modification/ combination classifiers' parameters. The servers and clients can modify the classifiers and then store them in their storage separately. Both personal classifiers and *standard classifiers* can be modified.

System contexts (such as privacy & policy rules)

Privacy rules:

This part creates some rules based on received contexts from clients to protect user against threats. For instance, if a client had a dangerous operation, new rules will be created to solve this problem by issuing rules.

Policy rules:

The policy rules are created in this part. For example, the server recognizes clusters in sensor network and selects a client as head of a cluster or selects a cluster for a client (sensor) and etc.

4- 4) Database Layer

The function of this layer is to store the necessary data. These are important data that must be stored such as the personal classifiers, the modified classifiers of people, Parameters and rules for modifying the classifiers, the modified classifiers depend on contexts, Privacy & policy rules and other data that can be useful in the system.

5. REFRENCES

[1] D. Nicklas, M. Grossmann, J. M´ınguez and M. Wieland, "Adding High-level Reasoning to Efficient Low-level Context Management: a Hybrid Approach" in 6rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)

[2] A. Messer, A. Kunjithapatham, Ph. Nguyen, P. Rathod, M. Sheshagiri, D. Cheng, S. Gibbs, "SeeNSearch: A Context Directed Search Facilitator for Home Entertainment Devices" in 6rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)

[3] R.Prasad, C. Moran, F.Choi, R.Meermeier, S.Saleem, C.Kao, D.Stallard, P.Natarajan, "Name Aware Speech-to-Speech Translation for English/Iraqi", Spoken Language Technology Workshop, 2008. SLT 2008. IEEE15-19 Dec. 2008 Page(s):249 – 252

[4] K.Hsieh, Y.Liu, C.Lai; J. Lee, " The support of software design patterns for streaming RPC on embedded multicore processors " Signal Processing Systems, 2008. SiPS 2008. IEEE Workshop on 8-10 Oct. 2008 Page(s):263 – 268

[5] Sh.Chetan, A.Ranganathan, and R.Campbell. "Towards fault tolerance pervasive computing." in IEEE Technology and Society Magazine, 24(1):38–44, Spring 2005.

[6] P.Hu, J.Indulska and R.Robinson, "An Autonomic Context Management System for Pervasive Computing" in 6rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)

[7] L. Cardelli and A. D. Gordon, "Mobile Ambients," in First International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '98) at ETAPS'98, Lissabon, Portugal, 1998, pp. 140- 155

[8] M.M. Olsen, N. Siegelmann-Danieli, H.T. Siegelmann, "Robust artificial life via artificial programmed death Artificial Intelligence" Volume 172, Issues 6-7, April 2008, Pages 884-898

[9] L.Little, P.Briggs, " Pervasive healthcare: the elderly perspective " in PETRA '09: Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments ,ACM June 2009 Article No. 71 , ISBN:978-1-60558-409-6

[10] B.Okeke, K. L. Eddie Law , " Multi-level clustering architecture and protocol designs for wireless sensor networks", WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet , ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) SESSION: Wireless sensor networks ISBN:978-963-9799-36-3

[11] G.Ngai, S.Chan, J.Cheung, W.Lau, " The TeeBoard: an education-friendly construction platform for e-textiles and wearable computing " in CHI '09: Proceedings of the 27th international conference on Human factors in computing systems ,pages 249-258, ISBN:978-1-60558-246-7

[12] A. K. Dey, "Understanding and using context." Personal and Ubiquitous Computing, vol. 5, no. 1, 2001.

[13] H. Yoo, M. Kim, O. "Kwon, Emotional index measurement method for context-aware service", Elsevier Expert Systems with Applications 38 (2011) pages:785–793

[14] E. Farran, A. Branson, B.King. "Visual search for basic emotional expressions in autism; impaired processing of anger, fear and sadness, but a typical happy face advantage" Elsevier Research in Autism Spectrum Disorders, Volume 5, Issue 1, January-March 2011, Pages 455-462

[15] P.C. Petrantonakis, L.J. Hadjileontiadis, "Emotion Recognition From EEG Using Higher Order Crossings " IEEE Transactions on Information Technology in Biomedicine, Volume: 14 , Issue: 2 ,Publication Year: 2010 , Page(s): 186 – 197

[16] L.Rabiner, B.H.Juang, "Fundometal of speech recognition" ISBN:978013151575, pages:69-139