Hands-free speech-sound interactions at home

Pierrick Milhorat, Istrate Dan, Jérôme Boudy, Gérard Chollet

HANDS-FREE SPEECH-SOUND INTERACTIONS AT HOME

P. Milhorat[1], D.Istrate[3], J. Boudy[2], G. Chollet[1]

[1]Télécom ParisTech, 37-39 rue Dareau, 75014, Paris, France
[2]Télécom SudParis, 9 rue Charles Fourier, 91011 Evry Cedex, France
[3]ESIGETEL, 1 Rue du Port de Valvins, 77210 Avon Cedex, France

ABSTRACT

This paper describes a hands-free speech/sound recognition system developed and evaluated in the framework of the CompanionAble European Project. The system is intended to work continuously on a distant wireless microphone and detect not only vocal commands but also everyday life sounds. The proposed architecture and the description of each module are outlined. In order to have good recognition and rejection rates, some constraints were defined for the user and the vocabulary was limited. First results are presented; currently project trials are underway.

Index Terms— speech recognition, sound processing, sound recognition, domotics.

1. INTRODUCTION

The CompanionAble European project aims at combining smart home functionalities with mobile robot abilities for dependent people. The robot is the front-end of the domotic system (turning on/off the lights, shutting/opening the curtains, playing/stopping music, etc) as well as an everyday helper. Supported by external sensors in the house (infra red sensors, door opening detectors, etc) and internal data (camera, sonar, etc), it's an assistant reacting to predefined scenarios (homecoming, video call, etc) or defined by the user himself (task reminder, pill dispenser, etc).
To achieve such variety of tasks, the device is equipped with a touch screen. A mobile tablet and a static screen on the kitchen wall are also available. These are the three means to access the common graphical user interface of the system.
Esigetel and the Mines-Télécom institute gave the robot its vocal interaction ability. A list of domotic commands have been extracted from practical experiments with end users.
Other applications, for instance the agenda, the cognitive training or the robot control are also accessed via vocal commands. In both cases, commands are not only words but full natural language sentences.
Lots of projects were about speech recognition; current commercial systems show us how the vocal interaction may be widely available in a near future. However, our work tries to solve the issues related to the distance to the microphone. In our configuration, we use a single microphone on top of the robot which can drive anywhere in the one-floor house. The noise environment is also unrestricted and traditional. Noise subtraction methods with dedicated microphone recording hypothetical noise sources are difficult to be applied to this real time changing environment.
The CompanionAble project is further detailed in the second part of this paper. Sections 3 and 4 are about the sound processing and classification process, then, in section 5, the speech recognition system is described. Section 6 presents the first evaluations. Conclusions and perspectives drawn from this work are presented in the final part.

2. COMPANIONABLE

CompanionAble stands for Integrated Cognitive Assistive & Domotic Companion Robotic Systems for Ability & Security. This project is funded by the European commission and is composed by 18 academic and industrial partners. Partners are from France, Germany, Spain, Austria, Belgium, the Netherlands and the United-Kingdom. The main objectives are:

- To combine mobile companion robot ability with smart home functionalities
- To support social connection for dependent people
- To improve the quality of life and the autonomy of elderly people

Esigetel and the Mines-Telecom institute are leaders, each, to develop a vocal interaction and a multimodal distress situation detector. They take part in the person localization within the house as well. This paper focuses on the acoustic work.
Currently, the project is tested by end users in SmH Eindhoven (Netherlands) and LabinHam in Gits (Belgium). They are invited to try the whole system for several consecutive days.

3. SOUND PROCESSING ARCHITECTURE

The sound is acquired continuously through two parallel systems: a first one which is able to detect and classify sound events between existing sound classes; another one

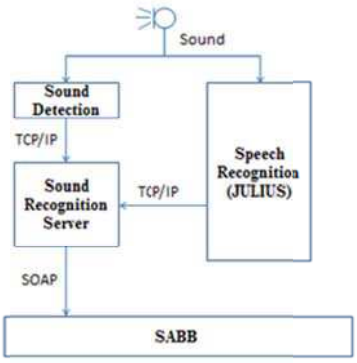

Figure 1 - Sound Processing Architecture

which is the speech recognition system. Figure 1 shows the communication between the sound and speech modules which is achieved using the TCP/IP protocol. The sound classification results and the vocal commands are sent to the CompanionAble architecture using SOAP protocol. The speech recognition output is filtered by the sound recognition system in order to avoid false alarms.

4. SOUND RECOGNITION

The sound recognition system is a two steps system: a detection module based on wavelet transform and a hierarchical recognition system (sound/speech and sound classification) based on GMM [2].

The sound classes used in CompanionAble trials were trained using sounds recorded with the CMT microphone (microphone under development from AKG) [3] in the Smart Homes (SmH) experimentation house. Currently the system has 5 sound classes: object fall, door ring, keys, cough and claps. The sound classes were chosen in order to help the CompanionAble system to identify distress situations and person activities.

The output of speech recognition is filtered by sound recognition module in order to avoid sending a wrong speech output instead of sound type identification. Because the two modules (sound and speech) are working in parallel, synchronization is needed. The sound module records in a buffer the three last sound/speech decisions associated with a timestamp. Then the decision of sending or rejecting the speech outputs within the server is effective between pairs of matching timestamps.

Each module was initially evaluated on pre-recorded data. For the sound the results for signals without noise was about 80% of good recognition rate [2]. The speech/no speech classification were experienced in SmH with a high rate near 95% of good classification.

5. SPEECH RECOGNITION

End-user difficulties, even inability to interact with the system using menus and touch screens justify the need for vocal interactions. Heavy cognitive or mobility troubles could make the system obsolete if it wasn't for the distant speech recognition. Three issues had been identified:

- Speech recognition in noisy uncontrolled environment
- Distant speech recognition
- Always-on speech recognition

Labs for practical experiments are based in the Netherlands and in Belgium, thus the human-machine interaction language is Dutch for the project.

Julius, developed by the Kawahara lab of Tokyo, was selected as the most appropriate recognition engine for a state-of-the-art speech recognition system [1]. It is able to process large-vocabulary search in real time, through a two-pass algorithm. It needs to be fed with N-gram language models and Hidden Markov Models (HMM) as acoustic models. The Julius engine can process the same audio input with several instances based on different language models.

The HMM of phones were trained on the *Corpus Gesproken Nederlands* (CGN). It is made of 800 hours of recorded and transcribed speech containing nearly 9 million words; this is the largest corpus for contemporary Dutch. Files are single speaker and multiple speakers recording, prompted or spontaneous speech.

Given the conditions (always on, distance to the microphone, uncontrolled noise environment) we present next some propositions to improve the robustness.

5.1. Trigger word

The dialog manager offered to lower the false positive rate with a trigger word. This word, when detected in the audio stream, increases the attention level which then decreases steadily according to time. When this level is positive, it triggers the recognition results analysis. For instance, the level is initially null: the speech recognition engine always processes the audio stream and outputs texts, as long as the trigger word can't be extracted from those segmented texts, transcriptions are rejected/ignored by the dialog manager. As soon as the attention level is above zero, the actual dialog starts and the manager analyses the received transcribed commands. While the dialog is sustained between the user and the system – either by repeating the trigger word or by evolution of the dialog – the attention level increases while silences, from the dialog manager point of view, decrease it. It can then reach its floor value and stop the input analysis. The selection of the attention word is important for the stability and liability of such a mechanism.

5.2. Speech/sounds classification

A speech recognition engine such as Julius search for the closest sequence of words matching the input audio observations given the probabilities contained in the acoustic model and in the language model. One may add a garbage model which will be the default match for unknown observation sequences or sub-sequences. In this application, every input is matched with a sequence of words. Thus sounds are processed as they were speech and a word sequence is returned. The sound classification prevents this to happen by discarding speech recognition results that occur while the stream has been classified as sound. It's a real-time process in parallel of the speech recognition one.

5.3. Acoustic adaptation

Acoustic adaptation methods have been studied at the earliest stage of the project [4]. Two adaptation methods were compared, namely: Maximum A Posteriori (MAP) and Maximum Likelihood Linear Regression (MLLR).
A language model was trained on a corpus of 57500 sentences derived from practical experiments and paraphrasing. The speaker is the same for the whole study, she has been previously recorded and the audio files are played through a loudspeaker. As expected, as only 10 phonetically balanced sentences are used, MLLR adaptation is the most suited technique. Without adaptation, 60% of the Julius' transcriptions are correct while this rate reach 70% with MAP adaptation and 73% with MLLR adaptation. Users go through MLLR adaptation before they use the system.

5.4. Language model combination

The first version of the speech recognition module was based on a single N-gram model trained on a 57658-sentence corpus. The acoustic model was adapted to fit the voice characteristics of the users using the MLLR method. This first system presented too much false positives, i.e. unwanted commands, when put to practical tests.
In order to improve both the recognition and rejection rates, a filter, described next, was implemented.
The dialog is based on frames [5]. These frames contain sub-dialogue graphs and transitions between states are triggered by the robot internal state/variables and the user inputs (vocal commands, buttons or/and sensors). A frame is enabled when at least one of its activation conditions is fulfilled; these are the same kinds of variables than the intra-frames ones. Thus one can build a dialogue hierarchy: the root frame which is initially enabled contains all the activation events to enable the sub frames and terminal states allow the sub frames to hand over the control to the main frame.
The sub frames have been clustered within eight classes. Each class lists all the vocal commands which are allowed and can be interpreted in the compound frames. A language model is build from those lists.
A 9^{th} language model is trained on the activation commands and is associated to the main frame.
Even while the speech recognition module doesn't receive information about the state of the dialogue, nine instances of the recognition engine run at the same time and deliver transcriptions of the input audio stream.
This language model selection process improves the good recognition rates for the application commands but on the other hand doesn't solve the rejection issues for out-of-application sentences.

5.5. Similarity test

Similarity between two recognizers' hypothesis is an extended Levenshtein distance. This is the total number of operation (substitution, deletion, insertion) to transform a sentence in another one. Furthermore it is normalized with the count of word in the sentences. Depending of the relative value of this distance, given a threshold, the hypothesis recognized by an engine fed with a specific language model is accepted or discarded. This test is used to:

- Confirm good recognition: a well recognized command according to both the general decoder and the specific decoder is validated. The exact specific decoder's hypothesis is sent
- Reject wrong hypothesis: a command recognized only by the general decoder is rejected.
- Correct partially correct hypothesis: a command recognized by a specific decoder while the general decoder outputs a close match is corrected: the specific hypothesis is sent

The general language model must, in this setup, recognize the sequences of word contained in the specific language models. One needs to add the whole set of commands in the training corpus of such a general model. We introduced a weight for these additional sentences which has been experimentally defined to be 1000: the commands were added 1000 times.
Finally, the test is not effective between one hypothesis for each decoder. We found out that it is better to use the n-best ones; it improves the good recognition rate:

- One hypothesis is outputted for specific decoder because of the size of their language model
- Several hypothesis (3 in our application) are produced by the general decoder and then fed to the similarity test

All this improvements were implemented. A first evaluation of those is presented next.

6. EVALUATIONS

A test corpus has been recorded in SmH with 5 speakers.

Each one of them spoke 58 sentences: 10 adaptation sentences, 20 application commands, 22 out-of-application sentences and 6 modified sentences. The modification is actually a deletion of one or several words in a command from the application.

The experiment setup is shown on Figure 2, Audio Dutch sequences are played through a loudspeaker and recorded by a microphone. The top loudspeaker was used for the experiment second phase and simulated noisy conditions. The sound level was of about 60 dBA which is the level of an average speaker standing about one meter away from the listener.

The first phase was intended to set the value of the commands 'weight in the general model and the number of hypothesis from the general recognition process for the similarity measure.

For vocal commands allowed by the application, the baseline, which is fed with a language model trained on all the commands, recognizes 15% of them.

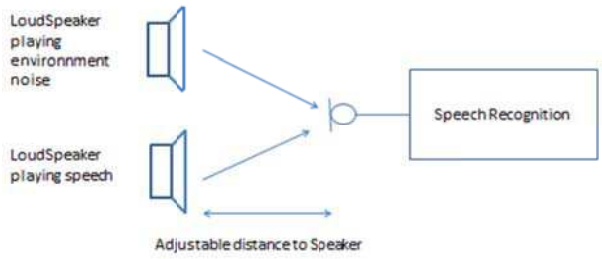

Figure 2 – Experimental setup for the tests

The similarity test addition improved these results up to 20% but one could notice an increase of the false-positive rate. The best system can recognizes commands with 85% of accuracy and never gives false-positive cases.

Every out-of-application sentences have been rejected by the system.

Modified commands are most of the time validated.

System	Recognition rate	False-positives rate
Baseline + adaptation	15	0
Baseline + adaptation + similarity test (commands' weight: 1; general decoder hypothesis: 1)	20	10
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 1)	55	0
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 3)	85	0

Table 1. Recognition rate and false-positive rate for in-application commands

System	Recognition rate	False-positives rate
Baseline + adaptation	15	0
Baseline + adaptation + similarity test (commands' weight: 1; general decoder hypothesis: 1)	20	10
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 1)	55	0
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 3)	85	0

Table 2. Recognition rate and false-positive rate for out-of-application commands

System	Recognition rate	False-positives rate
Baseline + adaptation	15	0
Baseline + adaptation + similarity test (commands' weight: 1; general decoder hypothesis: 1)	20	10
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 1)	55	0
Baseline + adaptation + similarity test (commands' weight: 1000; general decoder hypothesis: 3)	85	0

Table 3. Recognition rate and false-positive rate for mixed commands

In a second phase of the experimental process, the noise robustness was tested. The top loudspeaker plays ambient sounds.

As expected, the performances dropped. The good recognition rate is lowered as well as the rejection rate.

System	Recognition rate	False-positives rate
Wash machine	74	11
Dutch speaker	53	11
Music	47	5
Crowd	42	11

Table 4. Recognition rate and false-positive rate for in-application commands

System	Recognition rate	False-positives rate
Wash machine	0	0
Dutch speaker	0	0

Music	0	0
Crowd	0	3.64

Table 5. Recognition rate and false-positive rate for out-of-application commands

System	Recognition rate	False-positives rate
Wash machine	40	0
Dutch speaker	60	0
Music	20	0
Crowd	60	0

Table 6. Recognition rate and false-positive rate for mixed commands

7. CONCLUSIONS AND PESPECTIVES

The set up presented on this paper aims at providing spoken input for a companion robot within a smart home environment. As the robot is always on, so is the speech recognizer. Given these constraints, the most important characteristic to keep in mind is the robustness of the recognition. This combines both a good recognition rate but also accurate rejection criteria.

A trade-off between these two aspects has to be found. Is it acceptable to erroneously recognize a command? Can the user be asked to repeat utterances? During trials, it has been noticed that false positives could mean trouble and disturb the user. To solve this issue, the command/sentence set to be recognized has been restricted, this yields two other problems. Intended users are elderly and dependant people who get some trouble remembering specific commands. Furthermore, they could get quickly upset if the robot doesn't recognize their orders and think that this is useless, ignoring this functionality.

We proposed to experiment a combination of language models to improve the system accuracy.

A new general language model has been built from a read Dutch subset of the CGN corpus. Let's assume that it is able to recognize any Dutch utterances. Then another pass works on a restricted specific model with close vocabulary. The similarity between both resulting sentences, computed as a variation of the Levenshtein distance, behaves as a filter for acceptation/rejection.

A closer collaboration with the dialog manager would also bring more ways of refinement and filtering. The dialog manager of the CompanionAble project implemented in the companion robot follows a finite state automaton clustered in frames. Except for the root/main state which activates sub-frames and so is always active, we can select a specific language model built from acceptable sentences given a frame. Thus 10 restricted models have been created, one for each frame and one for the "main" frame. The dialog manager listens to the recognition process outputs, filtering them with what the current state(s) allow(s).

This more elaborated system proved to be robust enough to allow a good recognition rate as well as limited false positive cases. However, informal experiments showed its weakness when it comes to reject short commands, i.e. one-word sentences. The use of the robot's attention with the trigger word prevents this to happen.

8. ACKNOWLEDGMENTS

This work is supported by the FP7 European CompanionAble Project. We thank AKG in Vienna and SmartHome in Eindhoven for their assistance. We also thank Daniel Caon and Pierre Sendorek for their help in the first implementation of the speech recognition system.

9. REFERENCES

[1] A. Lee, "The Julius Book", http://julius.sourceforge.jp/juliusbook/en/, 2008

[2] J.E. Rougui, D. Istrate, W. Souidene, "Audio Sound Event Identification for distress situations and context awareness", EMBC2009, September 2-6, Minneapolis, USA, 2009, pp. 3501-3504

[3] J.E. Rougui, D. Istrate, W. Souidene, M. Opitz et M. Riemann, "Audio based surveillance for cognitive assistance using a CMT microphone within socially assistive technology", EMBC2009, September 2-6, Minneapolis, USA, 2009, pp.2547-2550

[4] D. Caon, T. Simonnet, J. Boudy and G. Chollet, "vAssist: The Virtual Interactive assistant for Daily Home-care", pHealth conference, 8th International Conference on Wearable Nano and Macro Technologies for Personalized Health. June 29th-Jult 1st 2011.Lyon, France

[5] S. Müller, C. Schroeter, H.-M. Gross, "Aspects of user specific dialog adaptation", International Scientific Colloquium, Ilmenau, Germany, 2010.