July 2013 Technical Report: KUL/ESAT/PSI/1303

Metadata for Corpora PATCOR and Domotica-2

Tessema N., Ons B., van de Loo J.,
Gemmeke J.F., De Pauw G.,
Daelemans W., Van hamme H.

PSI

Kasteelpark Arenberg 10 – box 2441
B-3001 Heverlee, Belgium

KU LEUVEN

Telephone: +32-(0)16-32.17.13 Fax: +32-(0)16-32.17.23 http://www.esat.kuleuven.be/psi

Technical Report(ALADIN-TR01)

Netsanet Tessema[1], *Bart Ons*[1], *Janneke van de Loo*[2], *Jort F. Gemmeke*[1], *Guy De Pauw*[2], *Walter Daelemans*[2], *Hugo Van hamme*[1]

[1]ESAT, KU Leuven, Leuven, Belgium
[2]CLiPS - Computational Linguistics Group, University of Antwerp, Antwerp, Belgium
bart.ons@esat.kuleuven.be, jort.gemmeke@esat.kuleuven.be

1. Introduction

In this report, we describe some technical aspects of the Adaptation and Learning for Assistive Domestic Vocal Interfaces (ALADIN) project. In this project, two speech corpora have been collected: **PATCOR**, more than two thousand spoken commands in the context of a card game, and **Domotica-2**, a corpus based on a home automation system that targets pathological speakers. Both corpora contain spoken commands in Belgian Dutch. Within the ALADIN project, we adopt a *frame*-based learning framework, i.e. we use semantic *frames* as a conceptual representation of the actions that are contained in a spoken utterance. A semantic *frame* is a data structure that contains all the relevant information associated with an action expressed in a spoken utterance. A *frame* contains one or more *slots*, and the entries of *slots* are called *slot values*. For every utterance spoken by a user, the ALADIN learning framework generates a *frame description*, which indicates which *slots* and *slot values* are present in the current utterance.

2. Datasets

In this section, we provide some background information to the *PATCOR* and *Domotica-2* datasets.

Table 1: Intelligiblity scores for the twenty participants of the Domotica-2 Dataset.

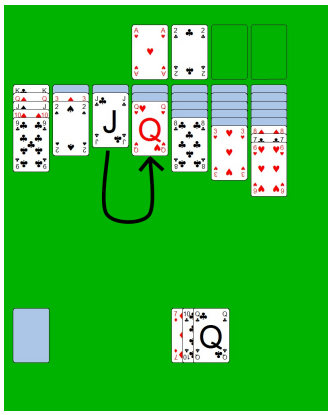
speaker id id	intelligibility score	speaker id	intelligibility score
28	73.14	11	94.26
29	73.59	40	85.50
30	68.99	41	64.22
31	73.72	42	68.77
32	65.64	43	89.35
33	66.16	44	89.20
34	76.20	45	83.17
35	72.32	46	74.86
37	69.40	47	73.41
17	88.57	48	85.77

2.1. PATCOR

The PATCOR corpus contains more than two thousand spoken commands in (Belgian) Dutch which have all been manually transcribed and annotated. During the recording sessions, the participants were asked to play the card game *patience* using only spoken commands, which were subsequently executed by the experimenter. The participants were asked to advance the game by using their own commands freely, in terms of vocabulary and grammatical constructs. A typical card game playing field scenario is shown in Fig. 1.

During data collection, each participant played in two separate sessions, with at least three weeks in between, so as to capture potential variation in command use over time. In total, we collected 2020 commands, averaging 253 commands per participant. The average number of moves per game is 55. For more information on PATCOR, please consult [1].

Put the jack of clubs on the queen of hearts



Frame	*Slot*	*slot value*
dealcard	<from_suit>	c
	<from_value>	11
	<from_foundation>	-
	<from_column>	-
	<from_hand>	-
	<target_suit>	h
	<target_value>	12
	<target_foundation>	-
	<target_column>	-

Figure 1: An example of a command, the associated action on the screen and the *frame description* of the action. A card is defined as the combination of a suit ((h)earts, (d)iamonds, (c)lubs, (s)pades) and a value from ace(1) to king(13). We also distinguish slots for "hands" at the bottom of the playing field, the seven columns in the center of the playing field and the four foundation stacks at the top right.

2.2. Domotica-2

This is the second part of the home automation datasets within the ALADIN project and specifically targets dysarthric speakers of Belgian Dutch. The Domotica-2 corpus was collected in two phases, taking into account the amount of effort it takes the pathological speakers to make recordings for the desired num-

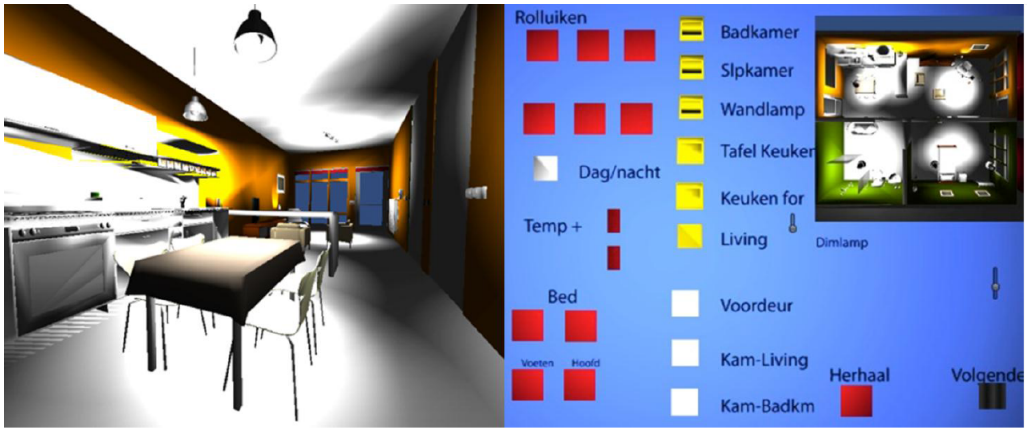


Figure 2: 3D house used in Domotica-2 data collection, left user's view, right wizard's view

ber of utterances per command.

In the first phase, nine pathological speakers were asked to control 31 devices in a simulated 3D environment, which is shown in Fig. 2. To provide some guidance to the test users, a simple story was read to them that described the envisioned scenario, alongside a visualisation of the scenario in the 3D environment. This ensured that the test users were not biased towards a particular choice of words or grammar when expressing the commands for the actions visualised in the 3D environment.

Nine different command lists were obtained for each of the 31 devices simulated in the 3D environment. To further give the test user a realistic computer interaction experience, the experiment was again performed in a Wizard-Of-Oz setup, in which the user could tour the 3D environment using the wizard interface shown Fig. 2, right) to generate the command lists.

In the second phase, 21 pathological speakers were recorded reading one of the nine command lists repeatedly in order to achieve the desired number of utterances. The resulting Domotica-2 database contains 2049 utterances, which were all manually transcribed and annotated. Eight out of 21 speakers have been diagnosed with Multiple Sclerosis, a progressive disease which leads to degeneration of one's voice.

For our experiments, one of the speakers is not used due to the small number of utterances recorded for this speaker. The intelligibility score of the remaining 20 speakers is given in Table 1. To calculate these scores, speech was recorded for each of the participants, while they were reading a short piece of text. The intelligibility scores were generated using an automated tool [2].

3. Semantic Frame Structure

In this section, we show how the semantic *frames* are integrated into a larger structure called *masterframes*. The *masterframes* contain all of the *frames* for a particular application and each *frame* contains all of the possible *slots* and each *slot* contains all of the possible *slot values*. We use *masterframes* in our implementation, so that we are able to generate the *frame description* of every utterance that might be spoken by a user in a particular scenario. The graphic representation of the *masterframes* for each of the databases is described in detail below.

3.1. PATCOR

The *masterframes* for PATCOR are shown in Fig. 3. In PATCOR, there are two types of *frames*, "Movecard" a *frame* describing the movement of a card on the playing field and "Dealcard" a *frame* that contains no *slots* or *slot values*, but simply triggers a new hand in the playing field. The playing scenario is shown in the Fig 1. The total number of *slot values* is 58. A typical utterance could be: *Put the jack of clubs on the red queen*. The generated *frame description* is shown in Fig. 1. The *slot value* pairs present in the this utterance are shown in the Fig. 1(below).

3.2. Domotica-2

The masterfranes for Domoica-2 are displayed in Fig. 4. In Domotica-2, we have four *masterframes*:

1. "increase heat" is used to increase the temperature of the room and does not have *slots* or *slot values*
2. "open/close" describes spoken commands that involve opening or closing of a domestic item
3. "triple commands" indicate spoken commands that deal with setting a domestic item into one of its three ranges (for instance the brightness level of a lamp)
4. "on/off" describes commands that involve turning on and off lights in the different rooms of the environment

The Domotica-2 database contains 27 *slot values*. A typical Domotica-2 utterance might be: *ALADIN, turn on the kitchen light*. The generated frame description for this command would be :

Frame	***Slot***	***Slot value***
on/off	<object>	kitchen_light
	<action>	on

4. slot value selection

The number of *slot values* used by every speaker, determines the complexity of the classification problem in the decoding process. In order to show this level of complexity, we have prepared a binary matrix indicating which *slot values* are used for each speaker in [3]. A 1 in the binary matrix indicates that a *slot value* is present for that user, while a 0 is used for a missing *slot value* in the data. Binary matrices for the PATCOR and Domotica-2 datasets are given in Fig. 5 and Fig. 6 respectively.

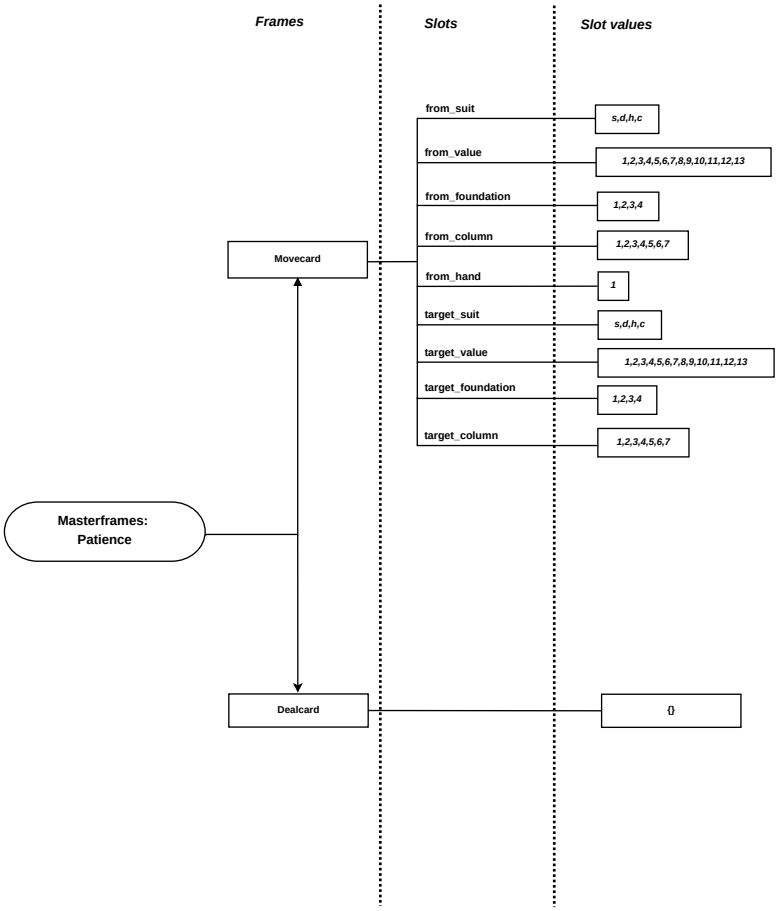


Figure 3: Masterframes for PATCOR, showing all the *frames*, *slots* and *slot values*. The "Dealcard" frame has no slots and the "Movecard" frame has eight slots. The *slots* beginning with "from" indicate from which position the card is moved and the *slots* beginning with "target" indicate to which position the card is moved.

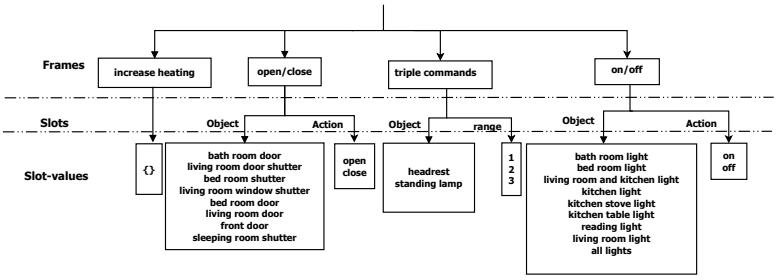


Figure 4: Masterframes in Domotica-2, showing all the *frames*, *slots* and *slot values*. Domotica-2 has four *frames*: "increase heating", "open/close", "triple commands", "on/off". The *slots* labeled "object" describe domestic objects (e.g door, window), while *slots* labeled "action" describe the actions applied to the domestic objects (e.g. turning an object on or off, closing or opening, ...)

Frame	slot	Slot-values	SPEAKER ID							
			1	2	3	4	5	6	7	8
Deal Card	{}	{}	1	1	1	1	1	1	1	1
Move Card	from suit	s	1	1	1	1	1	1	1	1
		d	1	1	1	1	1	1	1	1
		h	1	1	1	1	1	1	1	1
		c	1	1	1	1	1	1	1	1
	from value	1	1	1	1	1	0	1	1	1
		2	0	1	0	1	1	0	0	1
		3	1	1	1	0	0	0	0	1
		4	0	1	1	1	0	0	0	0
		5	1	1	0	1	1	1	1	0
		6	0	1	0	0	1	1	1	1
		7	1	1	0	0	1	0	1	0
		8	1	1	1	1	1	0	1	0
		9	1	1	0	1	1	1	1	0
		10	1	1	1	1	0	0	1	0
		11	1	1	1	1	1	0	1	1
		12	1	1	1	1	1	0	1	0
		13	1	1	0	0	0	0	0	0
	target suit	s	1	1	1	1	1	1	1	1
		d	1	1	1	1	1	1	1	1
		h	1	1	1	1	1	1	1	1
		c	1	1	1	1	1	1	1	1
	target value	1	1	1	1	1	0	1	1	1
		2	0	1	0	1	1	0	0	1
		3	1	1	1	0	0	0	0	1
		4	0	1	1	1	0	0	0	0
		5	1	1	0	1	1	1	1	0
		6	0	1	0	0	1	1	1	1
		7	1	1	0	0	1	0	1	0
		8	1	1	1	1	1	0	1	0
		9	1	1	0	1	1	1	1	0
		10	1	1	1	1	0	0	1	0
		11	1	1	1	1	1	0	1	1
		12	1	1	1	1	1	0	1	0
		13	1	1	0	0	0	0	0	0
	from foundation	1	0	1	0	0	0	0	0	0
	target foundation	1	0	1	0	0	0	0	0	0
	from hand	1	0	0	0	0	0	1	0	0

Figure 5: Binary Matrix for PATCOR. A 1 represents a *slot value* that is present for that speaker and a 0 represents a missing *slot value*. *slot values* which are not used by any of the speakers are not shown.

Frame	slot	Slot-values	Speaker ID																			
			28	29	30	31	32	33	34	35	37	17	11	40	41	42	43	44	45	46	47	48
Triple commands	object	headrest	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	0
	object	standing light	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	0
	action	1	1	0	1	1	1	1	0	0	0	1	1	1	1	1	0	1	0	1	1	0
	action	2	1	1	1	0	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0
	action	3	1	0	1	0	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0
On/off	object	bath room light	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	object	bed room light	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0
	object	living room kitchen light	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	object	kitchen light	0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	1	0
	object	Kitchen stove light	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	object	Kitchen table lamp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	object	reading light	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	object	living room light	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1
	object	All lights	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	action	on	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	action	off	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
Open/close	object	bath room door	1	0	0	1	1	0	0	0	1	1	1	1	1	1	0	1	0	1	1	0
	object	living room door shutter	1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0
	object	bed room shutter	1	0	1	0	1	0	0	0	1	1	1	1	1	0	0	1	0	1	1	0
	object	Living room window shutter	1	0	0	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0
	object	bed room door	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0
	object	Living room door	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
	object	front door	1	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0
	object	Sleeping room shutter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	action	open	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	action	close	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Increase Heating	{}	{}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Figure 6: Binary Matrix for Domotica-2. A 1 represents a *slot value* that is present for that speaker and a 0 represents a missing *slot value*.

5. References

[1] J. van de Loo, G. De Pauw, J. Gemmeke, P. Karsmakers, B. van den Broeck, W. Daelemans, and H. Van hamme, "Towards shallow grammar induction for an adaptive assistive vocal interface: a concept tagging approach," in *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, L. Rello and H. Saggion, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), 2012, pp. 27–34.

[2] C. Middag, "Automatic analysis of pathological speech," Ph.D. dissertation, Ghent University, Belgium, 2012.

[3] B. Ons, N. Tessema, J. van de Loo, J. Gemmeke, G. De Pauw, W. Daelemans, and H. Van hamme, "A self learning vocal interface for speech-impaired users," in *in Workshop on Speech and Language Processing for Assistive Technologies, SLPAT 2013*, 2013.