Learning Natural Language Understanding Systems from Unaligned Labels for Voice Command in Smart Homes

Anastasiia Mishakova, François Portet, Thierry Desot, Michel Vacher

Learning Natural Language Understanding Systems from Unaligned Labels for Voice Command in Smart Homes.

Anastasiia Mishakova, François Portet, Thierry Desot, Michel Vacher
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
anastasia.mishakova@gmail.com, {francois.portet,thierry.desot,michel.vacher}@univ-grenoble-alpes.fr

***Abstract*—Voice command smart home systems have become a target for the industry to provide more natural human computer interaction. To interpret voice command, systems must be able to extract the meaning from natural language; this task is called Natural Language Understanding (NLU). Modern NLU is based on statistical models which are trained on data. However, a current limitation of most NLU statistical models is the dependence on large amount of textual data aligned with target semantic labels. This is highly time-consuming. Moreover, they require training several separate models for predicting intents, slot-labels and slot-values. In this paper, we propose to use a sequence-to-sequence neural architecture to train NLU models which do not need aligned data and can jointly learn the intent, slot-label and slot-value prediction tasks. This approach has been evaluated both on a voice command dataset we acquired for the purpose of the study as well as on a publicly available dataset. The experiments show that a single model learned on unaligned data is competitive with state-of-the-art models which depend on aligned data.**

***Index Terms*—Natural Language Understanding, Smart Environments, Deep Neural Network, Voice-User Interface**

I. Introduction

Smart-homes with integrated voice-user interfaces (VUI) can provide in-home assistance to aging individuals [1], allowing them to retain autonomy [2]. It is also a domain of great interest for the industry [3]. Such systems usually include several modules, such as Automatic Speech Recognition (ASR), Natural Language Understanding (NLU) and Decision Making modules. The NLU module takes as input a transcript of the voice command provided by the ASR module and extracts its meaning in a form that can be processed by the Decision Making module.

To ease the interpretation of the utterance, voice command systems tend to impose a strict command syntax. However, studies on the interaction of elderly people with smart environments show that they were inclined not significantly deviate from the imposed grammar of the commands [2], [4], [5]. Among these deviations from the grammar, there were: misuse of keywords (name of the system that is supposed to trigger the NLU module), taking too long a pause within sentences, adding polite forms ("please", "could you..."), using the infinitive form instead of the imperative, ungrammatical sentences, using out-of-vocabulary words, etc.

This work is part of the VocADom project founded by the French National Research Agency (Agence Nationale de la Recherche) / ANR-16-CE33-0006.

Thus, the system that only processes well-formed commands does not seem flexible enough, which creates the need for a NLU system based on data rather than rules. However, statistical approaches used to rely on a high amount of aligned data such as in the BIO model [6] in which every single word of an utterance must be labeled as being part of a specific slot or not. Unfortunately, most new application domains will not have such a dataset available. In particular, in the domain of the smart home voice interaction, there is currently no such a dataset available which limits the development and reproducibility of the voice command system in smart environments. The closest datasets are either voice based but without voice commands [7] or designed for other tasks [8], [9].

To tackle this problem, in this work, we present an approach to learn NLU models from synthetic and unaligned data. Section II introduces the task, the sequence-to-sequence neural NLU model as well as the synthetic data generation method used to address the lack of data. In Section III, the collection of a multimodal dataset of voice interactions in a real smart home is briefly introduced. This dataset is used to evaluate the NLU models. This section also contains evaluation of seq2seq and the state-of-the-art NLU models trained on artificial data and evaluated on the collected data. Subsequent experiments show the influence of different input representations on the NLU performances. The paper ends with a short discussion and conclusion.

II. Natural Language Understanding Task and Method

A. NLU as slot-filling

One of the most popular ways of addressing the NLU problem is *slot-filling* which consists in extracting the overall *intent* of an utterance and identifying the most important elements called *slots*. The intent reflects the intention of the speaker while the slots can be defined as the entities and relations in utterance which are relevant for the given task [10]. For instance, in the utterance `"Turn on the lamp"`, the intent is to act on a device (`set_device`) while the details of this action are in the slots `action=turn_on="turn on"` and `device=light="lamp"`. Here, a slot is composed of its label (`action, device`), its normalized

value (`turn_on,light`), and the text associated to it (`"turn on", "lamp"`).

Typically, NLU systems treat intent recognition as a classification task over the whole sentence while slot labeling is addressed using a sequence labeling approach. Such NLU systems as Triangular Random Conditional Field (Tri-CRF) [11], attention LSTM RNN (attRNN) [6], and open source commercial tool RASA[1] were developed for this approach. All these systems need aligned data such as exemplified below:

```
"text": "Turn on the lamp",
"intent":"set_device",
"entities":[
{
"start":0,
"end":22,
"entity":"action"
"value":"TURN_ON"
"text":"Turn on"
},
...
],
...
```

Tri-CRF and attRNN predict the intent and slot-labels simultaneously while RASA requires training 2 separate models – one for the intent prediction, and – another for slot-label prediction. Moreover, neither Tri-CRF nor attRNN are able to learn slot-labels and slot values at the same time. Hence, two separate models are needed to perform label and value prediction.

In this paper, we propose one model to perform intent, slot-label and slot-value jointly on unaligned data. For instance, from the input `"Turn on the lamp"` the model should output a sequence like this `intent[set_device], action[TURN_ON], device[lamp]` which is sufficient for decision making.

B. Learning model

Classical seq2seq model architecture [12] has been studied on various NLP tasks including NLU [13]. A typical seq2seq model is divided into an encoder – which encodes the input sentence into fixed-length vector–, and a decoder – which decodes the vector into a sequence of words. Both the encoder and decoder are generally Recursive Recurrent Networks (RNN). This model is able to treat a sequence of words of variable size and has become the standard approach for many Natural Language Processing tasks. Briefly, a recurrent unit, at each step t takes an input x_t and a previous hidden state h_{t-1} in order to compute its hidden state and the output using:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h),$$

$$y_t = \sigma_y(W_y h_t + b_y),$$

where y_t is the output vector at each step; W, U, b are the parameters of the neural layer and σ_h and σ_y the activation functions of the neural layers. Once the encoder read the entire input sequence of words (i.e., it read the special token $< EOS >$), the last hidden state h_t is passed to the decoder which begins to output a sequence of words using the previous hidden state and the previous predicted vector as input (using the special $< SOS >$ token as trigger) until it generates the end of a sequence (i.e., $< EOS >$).

In most NLP tasks, to prevent the exploding/vanishing gradient problem and to model long dependencies in the sequence, Long Short-Term Memory (LSTM) or Gated recurrent units (GRUs) are used as basic units of RNN. Furthermore, to enable the decoder to base its prediction not only on the previous word and hidden state, but also on the hidden states of the input, the attention mechanism was introduced [14]. In that case, the decoder uses other information during the decoding which is the context vector c. At each step i and based on the input sequence length T_x:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij}\, h_j.$$

The weight α_{ij} is computed as follows:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

where e_{ij} is computed as follows:

$$e_{ij} = a(s_{i-1}, h_j).$$

e_{ij} represents an alignment or attention model that tells the decoder at step i which part of the hidden state of the input sequence to attend. The alignment model a can be a simple feed-forward neural network jointly trained with the rest of the architecture. The probability α_{ij}, reflects the importance of h_j with respect to the previous hidden state $i-1$ of the decoder in deciding the next state i and generating the output. Hence the decoder decides which parts of the source sentence to pay attention to. This is particularly useful when the next slot information to output depends on an input word far away in the input sequence.

Apart from bringing the ability to learn from unaligned data, the seq2seq model has two other very interesting features for our task.

1) seq2seq can be trained using two input representations: word and character. While word input representation is the most frequent, it generally fails to address unseen words in the input. Character-based input representation treats the input sequence character by character, and has been shown to be able to learn sub-word representations and to process unseen words [15], [16]. However, a model trained in the character mode would have to learn spelling of words which is computationally more expensive. Thus, the two representation modes are investigated in this work.
2) Since the decoder uses the first prediction as input to the second step of the decoding, we supposed that predicting the intent first will help predicting the following slots. It seems intuitive since each intent is associated with a

[1]https://rasa.ai/products/rasa-nlu/

certain set of slots. Therefore we assessed this hypothesis by training two separate seq2seq models: one on the dataset with slots only and another one on the dataset containing intents coupled with slots.

To be able to train seq2seq models we followed the methodology similar to the one used in [17]. Seq2seq models were trained on an artificially generated corpus and were tested on a real corpus. For the sake of comparison, we also trained and tested seq2seq models on the publicly available spoken dialog corpora PORTMEDIA in the domain of festival show booking. The corpora used for training and testing are described in the following sections.

C. Corpora

1) Artificial corpus: To solve the problem of the absence of French domain specific training data, an artificially annotated corpus was generated using a Feature Grammar class of Python's NLTK library [17]. Below is a training example: the sentence "`can you close the blind`":

```
"text": "KEYWORD tu peux fermer le store",
"intent": "set\_device"
"entities": [
  {
  "start": 16,
  "end": 22,
  "entity": "action",
  "value": "CLOSE",
  "text": "fermer",
  },
  {
  "start": 23,
  "end": 31,
  "entity": "device",
  "value": "blind",
  "text": "le store",
  }
],
}
```

The phrases of the dataset were generated from their corresponding semantic representations, each one containing an intent and one or more slots. For generating the phrase "`Turn on the light`" (`Allume la lumire`) which is defined by an intent `set\_device` and the slots `action` and `device`,a rule was created where `Dact\_set\_device` is the predicate, rewritten into the predicates defining the slots:

```
set_device[ACTION=?s,Location=?l, Device=?d] ->
    Slot_action[ACTION=?s], Slot_device[ALLOWABLE\_ACTION=?s,
                          Location=?l, Device=?d, ARTTYPE=?a]
```

Each predicate has attributes given in square brackets. For example, `Slot_action` and `Slot_device` has attributes `ACTION` and `ALLOWABLE_ACTION` which must have the same value (expressed by `?s`). This allows us to generate only meaningful commands, i.e. commands consisting of allowable actions applied to certain devices, for instance commands to turn on only devices that can be turned on. Each device has a list of allowable actions in the specification file of the grammar.

Each of the slots is further rewritten into words using the rules of the grammar.

As a result of this generation process, the artificial dataset contains 8 intents. The examples of the phrases below are translated from French; the frequencies of each intent are given in parenthesis:

`Contact`	*Call my daughter* (595)
`Set_device`	*Start the boiler* (17375)
`Set_device_group`	*Turn all lights of the kitchen on* (10475)
`Set_device_property`	*Increase the volume of the TV* (5250)
`Set_room_property`	*Raise the temperature in the room* (2640)
`Check_device`	*Is the radio on?* (1868)
`Check_device_group`	*Are the doors closed?* (3982)
`get_world_property`	*What is weather like?* (10)

To avoid generating nonsensical phrases like "Turn on the oven in the bathroom", the generation is constrained by the usual location of devices. Other location features such as floor location were added for variation. For instance, instead of just generating the sentence "Turn on the oven in the kitchen downstairs", "Turn on the oven in the kitchen upstairs" is also generated. Both sentences are meaningful.

2) Real data: Two other small datasets collected in realistic conditions were used: The VocADom@A4H and SWEET-HOME corpora. The SWEET-HOME corpus was collected in the smart home DOMUS, equipped with microphones for speech recording, sensors for providing information on the user's localization and activity [18]. The VocADom@A4H corpus was collected in the smart home Amiqual4Home [17] and will be described in detail in III-A. The corpora were annotated following the same annotation scheme as the artificial corpus.

Finally, we also used the PORTMEDIA corpus in our study. It is a dataset of telephone conversations collected from the ticket reservation service for the festival of Avignon in 2010 [19]. It contains 700 annotated dialogues. While this dataset is not related to smart home it is the only available in French of that size and quality. It will be used in the experiment as a benchmark to assess the genericity of the approaches.

Table I presents the statistics for all corpora.

TABLE I
COMPARISON OF THE USED CORPORA: ARTIFICIAL, VOCADOM@A4H, SWEET-HOME AND PORTMEDIA.

Parameters	Artif2	VocADom@A4H	Sweet-Home	PortMedia
phrases	42195	1646	727	18026
words	156	285	120	3062
characters	43	44	42	71
intents	8	6	7	4
slot-labels	16	11	7	32
slot-values	60	44	24	378

These two kinds of corpora used in this work - a big dataset of conversational real data PORTMEDIA and the artificial corpus show significant differences. PORTMEDIA contains only 4 types of intent, while the artificial corpus contained 7; therefore the task of intent prediction is more challenging in the case of the artificial corpus. However, regarding slots PORTMEDIA contains a richer set of labels and values making it more challenging on this respect.

For the need of the study, the dataset were converted into the sequence-to-sequence format. Below is a training example:

```
Input: "close the door"
Output: intent[set_device], action[close],
        device[door]
```

As it can be noticed in this example, the intent is included into the sequence of slot as first token. The intent values and slot values are included in square brackets. Slot-labels were separated from slot-values so that models can learn them separately. The intent was put as the first element in the sequence because it was supposed that generating the intent first will help to predict the following slots since slots distribution tend to depend on the intent.

D. Learning methods

The seq2seq model used in this paper is the attention-based encoder-decoder GRU Bidirectional RNN. The number of embedding units was 128, the number of encoder's and decoder's units was 128. Optimization algorithm was Adam at a learning rate of 0.0001. Batch size was 32. All these were default parameters of the seq2seq library that was used in the experiments[2].

As for the number of training steps and input/output sequence length, they are presented in the table II. The optimal input/output sequence size was calculated from the corpus. The training and test sets as well as the modes are specified in the same table.

TABLE II
OVERVIEW OF SEQ2SEQ MODEL LEARNING SCHEMES (W/I STANDS FOR WITHOUT INTENT", ARTIF REFERS TO ARTIFICIAL CORPUS)

No	Train	Test	mode	nb steps	I/O seq len
1	Artif+SWEETHOME	VocADom@A4H	word	150 000	50/100
2	PORTMEDIA	PORTMEDIA	word	150 000	150/50
3	Artif+SWEET-HOME	VocADom@A4H	char	50 000	100/150
4	PORTMEDIA w/i	PORTMEDIA w/i	word	150 000	150/50
5	PORTMEDIA w/i	PORTMEDIA w/i	char	50 000	300/100

To see how seq2seq models perform compared to the baseline, we will compare the models number 1, 2 and 3 from the table II with the baseline models. To see how seq2seq models perform in the word mode compared to the character mode, we are going to compare the models 1 and 3 between themselves. To see if the prediction of intent helps the prediction of slots, we trained models 4 and 5 on PORTMEDIA without intent (model 4 in the word mode, model 5 in the character mode) and we are going to compare them with seq2seq models trained on corpora with intent.

As for the baseline Tri-CRF, RASA and attRNN, there were two models for each of them: one trained on the artificial corpus and tested on VocADom@A4H, another trained and tested on PORTMEDIA. It is necessary to remind that each of these baseline models comprises 2 or 3 separate models for predicting intents, slot-labels and slot-values. Compared to three above-mentioned models – Tri-CRF, RASA and attRNN – seq2seq has as advantage that it does not require the alignment of slots of text segments, and it only requires training of one model for intent, slot-label and slot-value prediction.

[2]https://github.com/google/seq2seq

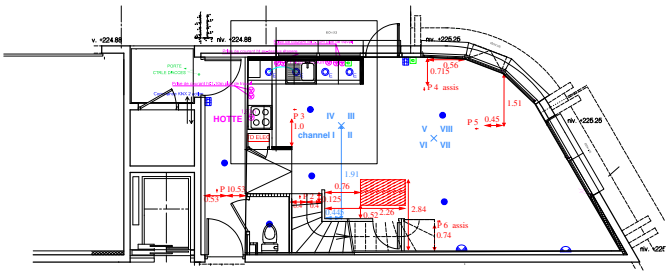

Fig. 1. Ground floor: kitchen and living room.

III. EXPERIMENT AND RESULTS

In this section, after a brief description of the collection of the realistic test dataset, the NLU experiments are reported.

A. VocADom@A4H dataset collection

In this work, the pilot smart home was the instrumented apartment of Amiqual4Home[3]. It serves as a realistic showroom for products and services for smart homes, and as a tool for user experiments [20]. This 87 m^2 Smart Home is equipped with home automation systems, multimedia controller, and means for observing human activity. The kitchen and the living room are on the ground floor (Fig. 1), the bedroom and the bathroom on the first floor. A dedicated hidden control room allows to centralize the recording of all the sensors and control all the devices remotely.

This Smart Home is fully functional and equipped with sensors, such as energy and water consumption, hygrometer and actuators able to control devices, lighting, shutters, multimedia diffusion, distributed in all rooms. Overall, it contains more than 500 controllable or observable items. In addition, 6 cameras are set up in the ceiling of the rooms. Home automation sensors and actuators (e. g., lighting, shutters, security systems, energy management, heating, etc.) are connected to a KNX[4] bus system (standard ISO/IEC 14543). Besides KNX, several field buses coexist, such as UPnP (Universal Plug and Play) for the multimedia distribution, X2D for the contact detection (doors, windows and cupboards), RFID for the interaction with tangible objects (not used in the VOCADOM project). The management of the home automation network, sending commands to the different actuators and receiving changes of sensor values, is operated through openHAB[5].

To build a realistic dataset, eleven participants were guided to performed activities of daily living for about an hour and to utter voice commands while doing so. The experiment was divided into three parts:

1) elicitation of voice commands: the participants were given images representing scenarios and had to guess how to utter commands to respect these scenarios.

[3]https://amiqual4home.inria.fr
[4]https://www.knx.org/
[5]https://www.openhab.org/

Images were chosen so that people were not lexically constrained and had thus to use their own words to express their intention.

2) multi-resident commands: a second participant entered the rooms and they both followed a scenario in which voice commands had to be uttered. They were not always in the same room.
3) background noise: the participants were given a list of phrases to read(e.g., *I lock the door, Vocadom turn down the light, téraphim turn off the radio in the bathroom*, ...) while a background noise was present (e.g., music, fan ...).

Data logging of all sensors and the control of the home automation system were performed from an hidden control room. Participants' voice commands were executed in a Wizard of Oz manner. The participants were not informed whether the system was automatic or not. In the following we refer to this resulting dataset as the VocADom@A4H dataset.

The audio part of the corpus was transcribed using Transcriber[6] while the semantic annotation of the voice commands for Natural Language Understanding (NLU) was performed using a web-based tool that was developed as part of our project. The total number of corpus commands was 1646 (not counting the sentences which were not commands).

It should be noted that while the SWEET-HOME corpus contains only read utterances, the VocADom@A4H contains spontaneous and read utterances.

B. learning results

The table III sums up the performances of the baseline models Tri-CRF, RASA and attRNN and seq2seq models on the VocADom@A4H corpus.

TABLE III
NLU MODELS F-MEASURE (%) PERFORMANCE ON VOCADOM@A4H DATASET (SH STANDS FOR SWEET-HOME)

Model	Corpus train	Intent	Slot-label	Slot-value
Tri-CRF	Artificial	85.84	**79.95**	63.27
Att-RNN	Artificial	**96.70**	74.27	**65.05**
RASA	Artificial_v1	76.57	79.03	61.95
seq2seq, word	Artificial + SH	*94.74*	51.06	34.95
seq2seq, char	Artificial + SH	72.31	*67.33*	*41.00*

The three baseline systems Tri-CRF, Att-RNN and RASA exhibit the best performances for all three tasks over the seq2seq models. However, the baseline systems were trained on aligned data while the seq2seq models were not. The seq2seq model using the *word* mode shows competitive intent classification performance but this task is not dependent on alignment. It is interesting to note that seq2seq using the *character* mode shows better performance that in the *word* mode for slot tasks.

To assess the performance of the models we also evaluated them on the PORTMEDIA corpus. The corpus was divided in a test set (10%), dev set (10%) and a train set (80%). This is reported on Table IV. Again the three baseline systems Tri-CRF, Att-RNN and RASA exhibit the best performances reaching about 95% for all three tasks. This shows that the models have been satisfactorily implemented and that the VocADom@A4H corpus is actually more difficult than the PORTMEDIA one according to the F-measure. This is also supported by the seq2seq using the *character* mode that performs better on PORTMEDIA than on VocADom@A4H but which is still far below the baseline.

TABLE IV
F-MEASURE (%) PERFORMANCE OF ALL SYSTEM ON PORTMEDIA TEST-SET (10% OF THE TOTAL)

Corpus train	Model	Intent	Slot-label	Slot-value
PORTMEDIA	Tri-CRF	96.36	95.39	92.32
PORTMEDIA	Att-RNN	**97.56**	**96.11**	**95.08**
PORTMEDIA	RASA	92.26	94.16	93.34
PORTMEDIA	seq2seq, word	97.08	64.21	58.06
PORTMEDIA w/i	seq2seq, word		*66.09*	*54.40*
PORTMEDIA w/i	seq2seq, char		63.42	53.22

IV. DISCUSSION

The first outcomes of the experiment is that all models show much better results on PORTMEDIA dataset than when trained on the artificial corpus and tested on the realistic corpus VocADom@A4H. This shows how difficult it is to account for the diversity of a realistic corpus compared to that of an artificial corpus. One possible reason is that the realistic corpus has been recorded with naive participants and that it contains significant variations of vocabulary and syntax with respect to the artificial corpus: repetitions, disfluencies and interjections (*euh*), keywords appearing at different positions etc. This results in utterances that are syntactically different from the artificial dataset. In addition, the vocabulary of the realistic corpus is bigger: 285 words for VocADom@A4H against 156 words for the artificial dataset leading to a high number of OOV words, with 142 words not occurring in the artificial dataset. A 3-gram language model learned on the artificial dataset shows a perplexity of 58 (without the $< s >$ tag) on the real corpus which is quite high for this task.

It should also be noted that seq2seq models are trained on the artificial corpus in addition with the SWEET-HOME corpus to ensure more lexical variability. Taking into account these differences, it is nevertheless interesting to compare the performance of seq2seq with state-of-the-art models. Seq2seq showed a performance equivalent to those of RASA, att-RNN and Tri-CRF, on the intent task but it did more poorly on the slot task. However, these 3 models use slot alignment with text segments of the input sentence, whereas seq2seq does not require it. Moreover, seq2seq uses only one model for all tasks while the other baselines uses from 2 to 3 models.

As for the different modes of seq2seq, we found that the performance of the word mode is better on the intents but it is worse on the slot-labels and slot-values, as compared to the character mode. This may be due to the fact that the character model is able to handle out-of-vocabulary words to a certain extend. It is worth noting that the character model must learn

[6]http://trans.sourceforge.net/

the spelling of each word, but while this can be prone to error, we did not observe much misspelled intentions or slots. We can compare our results to those of study [15], which uses seq2seq in the character mode on the corpus E2E [21], for the task of generation, but not for the automatic understanding task. On the positive side, they found that the model never hallucinated (i.e., it did not produce irrelevant slots) and produced very few repetitions. On the negative side, there were sometimes omissions. In our case, the model produced a lot of slot-value substitutions, for example, *action[lower]* instead of *action[turn_on]*.

Finally, while intent prediction is typically considered as a separate task from slot labeling, we also investigated the effects of predicting intention within the sequence of slots. The last two rows of the table IV show the results of predicting the sequence of slots only, without the intent information. It turns out that the presence of intention does not have much influence on the prediction of the slots. Thus, first, intent can be included in the slot sequence by keeping a great classification performance (hence no need for a specific intent classifier), second, this intent information in the sequence does not seem to have strong side effect on the slot labeling task.

V. Conclusion and further work

We showed that seq2seq models can be competitive compared to the NLU models that require the alignment between the segments of input phrases and semantic labels. Besides, our seq2seq models only requires one model for all three tasks - prediction of intent, slot-labels and slot-values. However, several problems can be pointed out that call for further work.

For all the models of the study - trained on PORTMEDIA or on the artificial dataset - the results of the prediction of slot-labels and slot-values were much lower than on intent. One reason for this may be the fact that the number of slot-labels and slot-values is much higher than the number of intents. In addition, some words with their corresponding slot-labels and slot-values have been under-represented in the artificial dataset (for example, the word *temperature* with the corresponding slot-label *room- property* and its value *temperature*). In this case, the models have shown near zero performance as they tend to predict more common slot-labels and slot-values. Hence one of the perspectives of the research is to ensure the equal distribution of all elements – intents, slot-labels and slot-values – during the learning.

Another perspective of the research is to collect more data in realistic conditions of a smart-home and to add it to the training dataset. Training the system on such a dataset would enable the NLU system to process more complex syntax and semantics.

References

[1] K. K. B. Peetoom, M. A. S. Lexis, M. Joore, C. D. Dirksen, and L. P. De Witte, "Literature review on monitoring technologies and their outcomes in independently living elderly people," *Disability and Rehabilitation. Assistive Technology*, vol. 10, no. 4, pp. 271–294, 2015.

[2] M. Vacher, S. Caffiau, F. Portet, B. Meillon, C. Roux, E. Elias, B. Lecouteux, and P. Chahuara, "Evaluation of a context-aware voice interface for ambient assisted living: qualitative user study vs. quantitative system evaluation," *ACM Transactions on Accessible Computing*, vol. 7, no. 2, pp. 5:1–5:36, 2015.

[3] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, ""your word is my command": Google search by voice: A case study," in *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*, A. Neustein, Ed. Springer US, 2010, pp. 61–90.

[4] S.-y. Takahashi, T. Morimoto, S. Maeda, and N. Tsuruta, "Dialogue Experiment for Elderly People in Home Health Care System," in *Text, Speech and Dialogue*, Brno, Czech Republic, 2003, pp. 418–423.

[5] S. Möller, F. Gödde, and M. Wolters, "Corpus analysis of spoken smart-home interactions with older users." in *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[6] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Interspeech 2016*, 2016.

[7] A. Fleury, M. Vacher, F. Portet, P. Chahuara, and N. Noury, "A french corpus of audio and multimodal interactions in a health smart home," *Journal on Multimodal User Interfaces*, vol. 7, no. 1, pp. 93–109, 2013.

[8] P. Chahuara, A. Fleury, F. Portet, and M. Vacher, "On-line Human Activity Recognition from Audio and Home Automation Sensors: comparison of sequential and non-sequential models in realistic Smart Homes," *Journal of ambient intelligence and smart environments*, vol. 8, no. 4, pp. 399–422, 2016.

[9] H. Choi, S. K.M., H. Yang, H. Jeon, I. Hwang, and J. Kim, "Self-learning architecture for natural language generation," in *Proceedings of the 11th International Conference on Natural Language Generation*, 2018, pp. 165–170.

[10] G. Tur and R. De Mori, *Spoken Language Understanding Systems for Extracting Semantic Information from Speech*. Wiley, 2011.

[11] M. Jeong and G. G. Lee, "Triangular-chain conditional random fields," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 7, pp. 1287–1302, 2008.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[13] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1442–1451.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[15] S. Agarwal and M. Dymetman, "A surprisingly effective out-of-the-box char2char model on the E2E NLG Challenge dataset," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 158–163.

[16] R. Qader, K. Jneid, F. Portet, and C. Labbe, "Generation of company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation," in *11th International Conference on Natural Language Generation*, Tilburg, The Netherlands, 2018.

[17] T. Desot, S. Raimondo, A. Mishakova, F. Portet, and M. Vacher, "Towards a French Smart-Home Voice Command Corpus: Design and NLU Experiments," in *21st International Conference on Text, Speech and Dialogue TSD 2018*, Brno, Czech Republic, 2018.

[18] M. Vacher, B. Lecouteux, P. Chahuara, F. Portet, B. Meillon, and N. Bonnefond, "The Sweet-Home speech and multimodal corpus for home automation interaction," in *9th Language Resources and Evaluation Conference (LREC)*, Reykjavik, Iceland, 2014, pp. 4499–4506.

[19] F. Lefèvre, D. Mostefa, L. Besacier, Y. Estève, M. Quignard, N. Camelin, B. Favre, B. Jabaian, and L. M. Rojas-Barahona, "Leveraging study of robustness and portability of spoken language understanding systems across languages and domains: the PORTMEDIA corpora," in *LREC*, 2012, pp. 1436–1442.

[20] P. Lago, F. Lang, C. Roncancio, C. Jiménez-Guarín, R. Mateescu, and N. Bonnefond, "The ContextAct@A4H real-life dataset of daily-living activities Activity recognition using model checking," in *CONTEXT*, vol. 10257, 2017, pp. 175–188.

[21] J. Novikova, O. Dušek, and V. Rieser, "The E2E dataset: New challenges for end-to-end generation," in *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany, 2017, pp. 201–206.