Towards End-to-End spoken intent recognition in smart home

Thierry Desot, François Portet, Michel Vacher

Towards End-to-End spoken intent recognition in smart home

Thierry Desot, Francois Portet, Michel Vacher

Towards End-to-End spoken intent recognition in smart home

1st Thierry Desot
GETALP Team
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG,
38000 Grenoble, France
Thierry.Desot@univ-grenoble-alpes.fr

2nd François Portet
GETALP Team
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG,
38000 Grenoble, France
Francois.Portet@imag.fr

3rd Michel Vacher
GETALP Team
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
Michel.Vacher@imag.fr

***Abstract*—Voice based interaction in a smart home has become a feature of many industrial products. These systems react to voice commands, whether it is for answering a question, providing music or turning on the lights. To be efficient, these systems must be able to extract the *intent* of the user from the voice command. Intent recognition from voice is typically performed through automatic speech recognition (ASR) and intent classification from the transcriptions in a pipeline. However, the errors accumulated at the ASR stage might severely impact the intent classifier. In this paper, we propose an End-to-End (E2E) model to perform intent classification directly from the raw speech input. The E2E approach is thus optimized for this specific task and avoids error propagation. Furthermore, prosodic aspects of the speech signal can be exploited by the E2E model for intent classification (e.g., question vs imperative voice). Experiments on a corpus of voice commands acquired in a real smart home reveal that the state-of-the art pipeline baseline is still superior to the E2E approach. However, using artificial data generation techniques we show that significant improvement to the E2E model can be brought to reach competitive performances. This opens the way to further research on E2E Spoken Language Understanding.**

Index Terms: spoken language understanding, automatic speech recognition, natural language understanding, ambient intelligence, voice-user interface

I. Introduction

Voice based interaction in a smart home has become a feature of many industrial products. To be efficient, these systems must be able to extract the *intent* of the user from the voice command. Intent recognition is a subtask of Spoken Language Understanding (SLU). Its aim is to extract the meaning contained in an utterance [1]. Voice based intent recognition is typically performed through automatic speech recognition (ASR) and intent classification from the transcriptions in a pipeline. However, the intent classifier is trained on clean transcriptions whereas ASR transcriptions contain errors reducing the overall performance. Although the pipeline approach is widely adopted, there is a rising interest for end-to-end (E2E) SLU which combines ASR and NLU in one model, avoiding the cumulative ASR and NLU errors of the pipeline approach [2], [3]. The main motivation for applying the E2E approach is that word by word recognition is not needed to infer intents. On top of that, the phoneme dictionary and language model (LM) of the ASR become optional. However, E2E approaches are highly dependent on large training data sets which are difficult to acquire, limiting the applicability to new domains where data is scarce which is the case for smart homes.

The main contributions of this paper are: 1) the first work on E2E SLU for voice command in a smart home environment; 2) a comparison of a state-of-the-art pipeline approach that predicts intents from the ASR hypothesis and an E2E SLU model; 3) experiments performed with realistic non-English and synthetic data to deal with the paucity of domain specific data sets. Both approaches are positioned with respect to the state-of-the-art in Section II and are outlined in Section III. We tackle the lack of domain-specific data by using Natural Language Generation (NLG) and text-to-speech (TTS) to generate French voice command training data. An overview of these processes and data sets is given in Sections III and IV. Section V presents the results of experiments on a corpus of real smart home voice commands followed by a discussion, conclusion and outlook on future work.

II. Related work

SLU is typically seen as a slot-filling task in order to predict the speaker's *intent* on the one side and entities in a spoken utterance (*slots* and *values*) on the other side [1]. The most common approach is a pipeline of an ASR and an NLU module. The ASR system outputs the hypothesis transcriptions from a speech utterance that are analyzed by the NLU module to extract the meaning. While the slot-filling task is most often

This work is part of the ANR VocADom project (ANR-16-CE33-0006) funded by the french Agence Nationale pour la Recherche.

978-1-7281-0984-8/19/$31.00

addressed as a sequence labelling task, intent recognition is generally approached as a classification task over the overall transcription.

To address the cascading error effect of classical pipeline SLU models, such approaches used confidence measures and N-best lists. For instance, weighted voting strategies combining ASR output confidence measures and N-best list hypotheses were used in a Named Entity Recognition (NER) task [4] to take uncertainty into account. Since the n^{th} hypothesis tends to contain more character errors than the n-1^{th} hypothesis, a named entity (NE) label is considered correct if it occurs in more than 30% of the n-best candidates. This brought an improvement over the baseline F-measure (1-best) with 1.7%. Another method is to learn NLU models on noisy ASR transcriptions. In [5], manual and ASR output transcriptions with word ASR confidence measures were used for a NER task, to learn a support vector machine-based (SVM) NER system. This increased precision by 2% as compared to the baseline.

More recently, to improve ASR error handling, acoustic word embeddings for ASR error detection were trained through a convolutional neural network (CNN) based ASR model to detect erroneous words. Output of this ASR model is fed to conditional random fields (CRF) and an attention-based RNN NLU model [6]. The CRF outperformed the RNN approach and the concept error rate (CER) decreased by 1% integrating confidence measures. Previous approaches of SLU especially focused on tuning the ASR model or using N-best hypotheses. [7] modified the ASR dictionary and language model to directly generate transcriptions with NE labels. This led to a significant increase of slot recognition.

Only recently some E2E work integrates deep neural networks (DNN): in [2], intents were directly inferred from audio MFCC features training a sequence-to-sequence (seq2seq) model on clean and noisy speech data. This gave an accuracy of 74.1% on an in-house corpus (35 types of intent), while a seq2seq NLU model fed with the ASR outputs gave 80.9%. A similar E2E approach was applied in [3]. The author trained the Baidu Deep Speech ASR system [8] on NE annotated transcriptions. The training set was increased by performing NER on a large speech data set. Their system exhibited a better identification of NER labels than a pipeline system (69 vs 65% F-measure) but was less performing with NE values extraction (47 vs 50% F-measure). This overview shows that E2E SLU models generate high expectations for joint ASR and NLU optimization but their performances have not superseded those of the pipeline approach yet. A common outcome is that data augmentation is the key factor for bringing the E2E model to superior performance. [9] used TTS to improve speech recognition. Gadde *et al.* used an ASR E2E convolutional NN model with connectionist temporal classification (CTC) and report optimal ASR performances with 50% synthetic and 50% natural speech data in the acoustic model [9]. This aspect supports the data augmentation strategy that is used in this paper which is developed in section V-B.

III. Intent Recognition from Speech: Pipeline and E2E SLU methods

A. Pipeline Intent Recognition

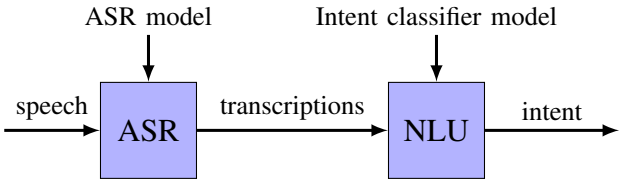

Fig. 1. Block Diagram of the Pipeline Intent Recognition Method.

The baseline pipeline method follows the diagram of Figure 1. It is composed of a first stage of ASR that extracts the transcription hypotheses from speech which are fed to an NLU module that selects the most probable intent from the hypotheses.

The ASR model is based on the ASR open source hybrid HMM-DNN Kaldi tool that uses speaker adapted features from the Gaussian mixture model (GMM) [10]. We used the nnet2 version which supports using multiple GPUs [11].

The Intent Classifier is a seq2seq attention-based PyTorch model. Recently, such models have been successfully used for the NLU slot-filling task [12]–[15] and supersede the previous state-of-the-art CRF model [16]. An important factor that explains the improvement of NLU models (including the CRF ones) is the application of *multitask learning*. Intent recognition is performed jointly with slot recognition [14], [16] which boosts performances for both tasks. Hence, many intent classifiers are trained within a framework that considers both tasks together. For that reason we propose a seq2seq model that encodes the sequence of words and decodes a sequence of symbols representing the global intent and each slot contained in the sequence to support the intent classification. For the example utterance *"Turn on the light"* the model generates the sequence `intent[set_device], action[TURN_ON], device[light]`. In this case, the intent is to `set` a device and the slots `action` and `device` provide information about which entities are concerned with the voice command.

The approach we propose has several advantages. First, contrary to most NLU methods that approach slot-filling as a *sequence labelling* task, we define the problem as a *generation* task. State-of-the-art approaches depend on *aligned data*. A sequence labelling task requires that each word in the transcription is assigned one unique slot label (e.g., the *BIO* NE labeling scheme). However, since our ultimate aim is to extract the intent directly from the raw speech signal, a sequence labelling approach is not adequate. It would require either to label each word in every n-best ASR hypothesis or to annotate each speech frame with a slot label. Our approach does not need aligned data and is thus more adapted to E2E intent classification from speech than the sequence labelling one.

The intent classifier we propose is close to the one of Liu *et al.* [14]. Both classifiers have shown close performances on a voice command task [17], [18]. Although the classifier of Liu *et al.* [14] has shown slightly better performances, it relies on aligned data while our intent classifier is independent from aligned data. Furthermore, since ASR errors reduce the performance of the NLU model, using unaligned data provides the flexibility to infer slot labels and values from imperfect transcriptions in order to recognize the intent. In summary, the ASR (Kaldi based) and the intent classifier (seq2seq) components represent together a strong pipeline baseline.

B. E2E SLU

The E2E approach is based on ESPnet [19]. It integrates the KALDI data preparation, extracts Mel filter-bank features and combines Chainer and PyTorch deep learning tools [20], [21]. The default PyTorch encoder is a pyramidal subsampling bi-LSTM [22], whereas the chainer back-end supports CNNs. Mapping from acoustic features to character sequences is performed by a trade-off *hybrid* multitask learning that combines CTC [23] and an attention-based encoder-decoder. As the attention mechanism alone allows too flexible alignments, CTC guides attention alignment to be monotonic.

$$\begin{aligned} logp^{hyb}(y_n|y_{1:n-1}, h_{1:T'}) = \alpha\, logp^{ctc}(y_n|y_{1:n-1}, h_{1:T'}) \\ + (1-\alpha)\, logp^{att}(y_n|y_{1:n-1}, h_{1:T'}), \end{aligned} \tag{1}$$

where y_n is a hypothesis of output label at position n given $y_{1:n-1}$ and the encoder output $h_{1:T'}$. The score combination ($logp^{hyb}$) for the hybrid CTC/attention architecture, with attention p^{att} and CTC p^{ctc} log probabilities is performed during beam search. The weight α can be set manually in order to give more importance to attention or CTC. To leverage a possible text corpus, a character RNN language model can be provided for the decoding. The log probability p^{lm} of the RNN LM can be fused with the CTC attention hybrid output by:

$$\begin{aligned} logp(y_n|y_{1:n-1}, h_{1:T'}) = logp^{hyb}(y_n|y_{1:n-1}, h_{1:T'}) \\ + \beta\, logp^{lm}(y_n|y_{1:n-1}). \end{aligned} \tag{2}$$

Since ESPnet models the ASR task at the character level, our approach to predict intents from the input signal was inspired by [2] and [3]. The output target of the ESPnet process was speech transcriptions augmented with characters (e.g., `@`, `#`...) symbolizing the intent of the utterance. Hence, the ESPnet model is trained to predict *enriched* transcriptions where each hypothesis is contextualized by its global intent. This task is described in section V-B.

IV. Data Collection and Augmentation

The pipeline and E2E intent recognition methods described above have been applied to the case of voice commands in a smart home. The application context is illustrated Figure 2. Each time a dweller utters a command, this utterance is captured and analyzed by an SLU module. If the intent is to control the house (in the example, to turn on the light), the semantics extracted from the utterance are sent to the home automation system. Otherwise, the utterance is ignored. Hence, the intent recognition information is of primary importance for the decision-making module to make the home automation system activate a command or not.

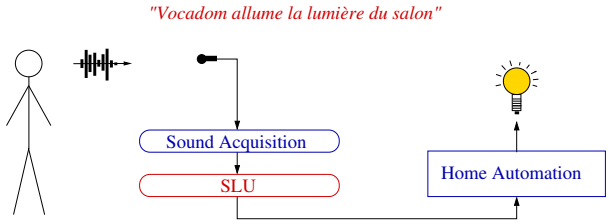

Fig. 2. Activation of the light by the Home Automation system after SLU analysis and intent recognition (*VocADom turn on the light in the living-room*).

Although voice based commands is a spreading feature of many IoT devices, there is a lack of speech based domain-specific corpora, especially for non-English languages such as French. To this end we collected a corpus in a real smart home with several users that is made available to the community [1]. Despite this corpus, the amount of data is far too low to train a DNN. For that reason we tackled this data scarcity problem using data generation. The next subsections outline the *realistic* data corpora available and the *artificial* training data generation.

A. Realistic Smart home data sets

Few French real domain-specific corpora are available. One can cite the SWEET-HOME corpus [24] which was recorded by participants enacting activities of daily living in a smart home equipped with home automation sensors and actuators. Continuous speech was mainly composed of voice commands. However it was recorded with only single user settings with a simple set of commands respecting a strict grammar and it is not sufficient to cover a large set of intents with a lot of syntactic and lexical variation.

Hence, we also used the VocADom@A4H corpus [25] which includes about twelve hours of audio signal and was acquired in realistic conditions in the two-storey Amiqual4Home smart home [2] (Fig. 3, 4 and 5). This 87m^2 smart-home with a kitchen, living room, bedroom and bathroom, is equipped with home automation systems, multimedia devices, and microphone arrays. More than 150 sensors and actuators were set in the house to acquire speech, control light, set the heating etc. Eleven participants uttered voice commands while performing activities of daily living for about one hour. Out-of-sight experimenters reacted to participants' voice commands following a wizard-of-Oz strategy to add naturalness to the corpus. The resulting speech data was semi-automatically transcribed, then humanly double-checked and resulted in 6,747 utterances, annotated with 8 different intents, including the `none` intent (for sentences without intent) and slot labels.

[1] https://vocadom.imag.fr
[2] https://amiqual4home.inria.fr

Fig. 3. Instrumented kitchen.

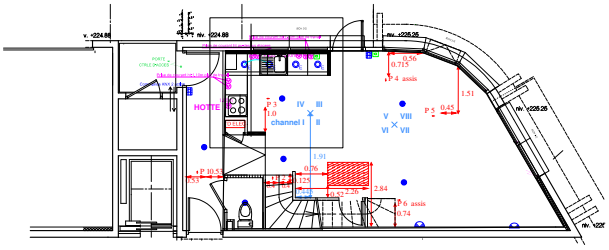

Fig. 4. Ground floor: kitchen and living room.

Fourteen different slot labels were defined such as the `action` to perform, the `device` to act on, the `location` of the device or action, the `person` or `organization` to be contacted, a device `component`, a device `setting` and the `property` of a location, device, or world. This corpus has been annotated by three annotators. Table I provides representative examples of voice commands with intent, slot and slot value labels. For each utterance the global intent is given with the slots between brackets. For the example *"are the lights upstairs on"*, `CHECK_DEVICE` means that the lights (`DEVICE`) on the upper floor (`LOCATION-FLOOR`) should be checked whether they are on (`DEVICE-SETTING`) or not.

TABLE I
EXAMPLES OF NLU ANNOTATED VOICE COMMANDS

Sentence + NLU annotation

are the lights upstairs on?
`CHECK_DEVICE(DEVICE=light="lights",`
`LOCATION-FLOOR=1="upstairs",`
`DEVICE-SETTING=on="on")`

call the doctor `CONTACT(PERSON-OCCUPATION=doctor="doctor")`
what time is it? `GET_WORLD_PROPERTY(WORLD_PROPERTY=time="time")`

open the blind
`SET_DEVICE(ACTION=open="open",`
`DEVICE=blind="blind")`

increase the volume of the radio
`SET_DEVICE_PROPERTY(ACTION=turn_up="increase",`
`DEVICE-COMPONENT=volume="the volume",`
`DEVICE=radio="of the radio")`

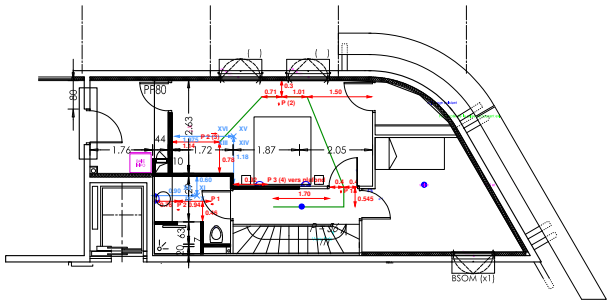

Fig. 5. First floor: bedroom and bath room.

B. Data augmentation via artificial data generation

Since the amount of real data is too small for training, the corpus generator of Desot *et al.* [17] was used to produce training data automatically labeled with intents, slot and value labels for the SLU experiments. On top of that several syntactic variants per sentence are provided (table II). It was built using the open source NLTK python library to which feature-respecting top-down grammar generation was added. Semantic constraints prohibit the production of nonsensical utterances. In each produced voice command a keyword is used to activate the Smart Home. Most keywords (such as "Ichefix") are proper nouns of at least 3 syllables long to enable sufficient duration for detection. *"Ichefix call a doctor"* activates the Home Automation system whereas *"Call a doctor"* should not trigger any reaction. With this generator more than 77k voice commands were produced for training purpose. A complete overview of intents is presented in table III. As shown in this overview, the data set is imbalanced.

Although the current trend for data augmentation is to use constrained RNN language models [26], such systems still need a set of initial sentences for bootstrapping and are difficult to control and to make them generalize to unseen concepts. This is why standard expert-based NLG was used in this work [27].

Finally, the ESLO2 corpus utterances (126h) of conversational French speech [28] was considered in the study. This corpus does not contain any voice commands but shares similarities with VocADom@A4H since it contains frequent disfluencies, repetitions, revisions and restarts [29]. ESLO2 was used to model the *none* intents in the training set. To extract a set of *none* intent utterances, an n-gram model was learned on the artificial corpus. Every utterance too close to the n-gram model was detected as a command-related utterance. All the detected sentences containing a token in a predefined list of domestic-related tokens were manually checked and put aside if they truly contained a domestic-related intent: e.g., *"You should open the door"* was set aside as it is actually a `Set_device` intent.

C. Summary of the data sets

Table IV summarizes the statistics for all corpora. The ESLO2 set is the largest one, without voice commands. The

TABLE II
EXAMPLES OF SYNTACTIC VARIATION WITH ANNOTATION IN THE ARTIFICIAL CORPUS

Sentence (French)	**English translation**
Ouvre la fenêtre	*Open the window*
Syntactic variation	
Ouvre la fenêtre s'il vous plaît	*Open the window please*
Est-ce que tu peux ouvrir la fenêtre?	*Can you open the window?*
Est-ce que tu peux ouvrir la fenêtre s'il vous plaît?	*Can you open the window please?*
Je veux que tu ouvres la fenêtre	*I want you to open the window*
Annotation	
`SET_DEVICE(ACTION=open="open",DEVICE=window="window")`	

TABLE III
ARTIFICIAL CORPUS (ARTIF.) AND VOCADOM@A4H (REAL.): EXAMPLES AND FREQUENCY OF INTENTS

Intent	**Example (French)**	**English translation**	**Frequency**	
			Artif.	**Real.**
`Contact`	*Appelle un médecin*	*Call a doctor*	567	114
`Set_device`	*Ouvre la fenêtre*	*Open the window*	63,288	2178
`Set_device_property`	*Diminue le volume de la télé*	*Decrease the TV volume*	7290	9
`Set_room_property`	*Diminue la température*	*Decrease the temperature*	3564	21
`Check_device`	*Est-ce que la fenêtre est ouverte?*	*Is the window open?*	2754	284
`Get_room_property`	*Quelle est la température?*	*What's the temperature?*	9	3
`Get_world_property`	*Quelle heure est-il?*	*What's the time?*	9	3
`None`	*La fenêtre est ouverte*	*The window is open*	-	4135

artificial data set contains the highest frequency of voice commands but has a smaller vocabulary than the VocADom@A4H corpus. The SWEET-HOME data set has a smaller size and less syntactic variation than the VocADom@A4H corpus that was used as *test* corpus in all the experiments (unless otherwise specified).

TABLE IV
COMPARISON OF THE CORPORA USED FOR NLU

Parameters	**ESLO2**	**Artificial**	**SWEET-HOME**	**VocADom@A4H**
utterances	161,699	77,481	1412	6747
vocabulary	29,149	187	480	1462
intents	1	7	6	8
slot labels	-	17	7	14
slot values	-	69	28	60

V. EXPERIMENTS AND RESULTS

A. Pipeline Intent Recognition Baseline Approach

Baseline ASR transcriptions were generated using Kaldi. We compared two acoustic models. The first one was trained on 90% randomly selected speakers of the corpora ESTER1 (100h) and 2 (100h), REPERE (60h), ETAPE (30h), SWEET-HOME (2.5h), BREF120 (120h) [30], VOIX-DETRESSE (0.5h) [31] and CIRDOSET (2h) [32]. For the second one we added 90% of the speakers of 126 hours of ESLO2 speech data [28], 10% being kept as development (DEV) set. The ASR dictionary consisted of 305k phonetic transcriptions of words based on the BD-LEX lexicon [33] to which phonetic variants were added with the LIA grapheme-to-phoneme conversion tool *LIA_Phon*[3]. For decoding, we used a 3-gram LM, based on the artificial corpus combined with the SWEET-HOME corpus. A generic LM was trained on 3,323M words, using EU bookshop, TED2013, Wit3, GlobalVoices, Gigaword, Europarl-v7, MultiUN, OpenSubtitles2016, DGT, News Commentary, News WMT, LeMonde, Trames, Wikipedia and our training data. The final LM resulted from an interpolation of the specific LM (weight = 0.6) with the generic LM (weight = 0.4).

The acoustic features are MFCC and were used to train a speaker-dependent triphone GMM model with speaker adapted transformation linear maximum likelihood regression (SAT+fMLLR). The final model was a hybrid HMM-DNN, mapping the transformed fMLLR characteristics to the corresponding HMM states. Word error rates (WER) in table V show that the fMLLR and HMM-DNN models with the ESLO2 data slightly outperform the acoustic models without it.

TABLE V
ASR PERFORMANCE (WER %) ON VOCADOM@A4H TEST

Model	**VocADom@A4H**
SAT+fMLLR	29.44
SAT+fMLLR (ESLO2)	27.99
HMM-DNN	23.3
HMM-DNN (ESLO2)	**22.9**

The NLU seq2seq model was composed of a bi-directional LSTM encoder and decoder. The input words were first passed to a 300-unit embedding layer. The encoder and decoder were each a single layer of 500 units. Adam optimizer was used with a batch size of 10, using gradient clipping at a norm of 2.0. Dropout was set to 0.2 and training continued for 10,000 steps with a learning rate of 0.0001. Input sequence length was set to 50 and output sequence length to 20. Beam search of size 4 was used. The NLU model was implemented using

[3]http://lia.univ-avignon.fr

the LIG PyTorch seq2seq library[4]. The training data was 90% of the combined artificial and the filtered ESLO2 data, the remaining 10% being the DEV set. The test data was the VocADom@A4H corpus. Both sets are described in section IV. F1-score at intent level on VocADom@A4H is shown in table VI. Results analysis shows a strong tendency towards `none` intent predictions due to the majority `none` intent class (unweighted manual).

We handled this imbalanced data problem by modifying the weight assignment in the cross entropy loss function of the PyTorch Seq2seq model. This was calculated on the complete training data and the resulting class weights were summed per batch. The total sum was multiplied with the cross entropy loss calculated per batch following equation (3):

$$weight\_class\_i = \frac{total\_instances}{instances\_class\_i} \quad (3)$$

The loss for majority intent classes dropped faster as compared to the loss for intent classes less represented in the training data. Consequently training increased for the minority intent classes. This method clearly improved performances (weighted manual).

NLU performances for intent predictions on the VocADom@A4H ASR output (weighted ASR) are worse than for the manual transcription predictions (weighted manual).

TABLE VI
INTENT CLASSIFICATION F1-SCORE (%) PERFORMANCES ON VOCADOM@A4H

Model	Intent
unweighted manual	76.95
weighted manual	**85.51**
weighted ASR	**84.21**

B. End-to-End Approach for Intent Recognition

For the E2E experiments, we used ESPnet default settings. The encoder was a very deep convolutional neural network (VGG) followed by six bidirectional (BLSTM) layers with 320 units. The decoder was a single LSTM layer with 300 units. The attention-CTC multi-task learning weight was set to 0.5. The optimizer was Adadelta with a batch size of 30. Training continued for 20 epochs. Beam size of 20 was used for decoding.

In this section, we describe the performance of ESPnet on a typical ASR task followed by the E2E intent prediction using an *enriched* transcription approach. ESPnet was first trained for an ASR task using the same training set as the Kaldi model in the pipeline approach (section V-A) and evaluated on the VocADom@A4H data set. The results reported in table VII show that ESPnet exhibits a higher Word Error Rate and Character Error Rate (CER) (real_data) as compared to Kaldi. However, when using the same LM data as in the Kaldi set-up (section V-A) for training and applying the character-based LM with ESPnet, the WER and character error rate (CER) improved (real_data+LM).

[4]https://gricad-gitlab.univ-grenoble-alpes.fr/getalp/seq2seqpytorch

Addition of synthetic speech data for an ASR task has proven to be beneficial to the ASR performance [9]. To compensate the lack of a large amount of domain-specific speech training data, the ASR training set described in Section V-A was augmented with TTS data generated on the complete artificial corpus using the open source French female SVOX voice[5] and represents 14.67% of the total acoustic model data. As shown in the third raw (real_data+LM+TTS), the addition of the TTS generated data brought significant improvement.

TABLE VII
ESPNET ASR WER (%) AND CER (%) ON VOCADOM@A4H

ESPnet training set	WER	CER
real_data	53.5	26.4
real_data+LM	50.6	23.9
real_data+LM+TTS	**46.5**	**22.9**

Although far from perfect, the results obtained on our DEV set (25.7% WER) are comparable to those obtained by Ghannay *et al.* [3] on their DEV set (20.70% WER) using the Baidu Deep Speech E2E ASR system. Moreover, Ghannay *et al.* [3] used a real corpus of newswire with similar conditions in training and test data, while in this paper, we deal with noisy domestic speech in the test data that is not present in our training set.

To perform intent recognition using ESPnet, we added intent labels (symbols) in the manual transcriptions of the corpus in sentence initial and final positions as follows:

`set_device`: "@ VOCADOM switch on the light please @".

The other symbolic labels per intent class are, '_' (`set_device_property`), '&' (`set_room_property`), '#' (`check_device`), ']' (`get_world_property`), '{' (`get_room_property`), '[' (`contact`).

For `none` intent sentences without voice command, no symbol was inserted. To study the impact of the synthetic data, different proportions of TTS data were used. For the creation of the character-based LM, we added the artificial corpus data with the intents injected as symbols into the data of the LM used with Kaldi in section V-A.

Table VIII mentions the hours of combined real and TTS training data (+tts) per model (TRAIN) and the percentage of the number of hours of TTS generated data in the acoustic model (SYNTH in TRAIN). Intent classes are not well predicted for the VocADom@A4H test set (+tts). These results pinpoint a too large distance between the acoustic features of the TTS data, and the VocADom@A4H natural speech data. This seems confirmed as performances increased when moving 1k sentences from the test set to the training set. Analysis showed that intent class prediction benefits more from the SWEET-HOME real data and the 1k test sentences added to the acoustic model, combined with the TTS data (+tts+VocADom@A4H_1k).

[5]https://launchpad.net/ubuntu/+source/svox

TABLE VIII
E2E INTENT RECOGNITION F1-SCORE (%) WITH ESPNET ON VOCADOM@A4H

Training set	TRAIN (hours)	SYNTH in TRAIN (%)	F1 TEST
+tts	553.9	14.67	47.31
+tts+VocADom@A4H_1k	554.5	14.41	50.99
+tts+VocADom@A4H_1k+inc	669.66	29.13	53.15
+tts+VocADom@A4H_1k+inc+LM	669.66	29.13	67.95
+tts+VocADom@A4H_1k+dec	84.69	94.39	53.92
+tts+VocADom@A4H_1k+dec+LM	84.69	94.39	**70.21**

Since the none intent class is over-represented, we handled imbalanced data in two ways: by decreasing the `none` intent class instances and by increasing instances of the underrepresented intent classes `set_device_property`, `set_room_property`, `check_device`, `get_world_property`, `get_room_property`, to about 20k instances per class which increased the F1-score (+tts+VocADom@A4H_1k+inc). Reducing the impact of the utterances without voice-command, by leaving only 11k utterances with a `none` class label in the acoustic model, slightly improved performance (+tts+VocADom@A4H_1k+dec). Decoding with the character-based LM including artificial corpus data augmented with the symbolic intent class labels, using our two best models, significantly increased the F1-score (+tts+VocADom@A4H_1k+inc+LM, +tts+VocADom@A4H_1k+dec+LM). The maximal E2E SLU performance was reached using an attention-CTC multi-task learning weight of 0.5.

For a fair comparison we also retrained the pipeline SLU ASR and NLU modules using the same reduced training data. Table IX (E2E SLU) recalls the best E2E performance from table VIII (+tts+VocADom@A4H_1k+dec+LM) and compares intent classification performance with he pipeline SLU model, trained on the reduced data set. With such a small training data set the E2E model is able to supersede the baseline pipeline approach for intent prediction. However this time the pipeline ASR (Kaldi) exhibited a WER of >90%. With an E2E ASR (ESPnet) training on the same reduced data a WER of 60.6% was obtained. Hence we used the resulting ESPnet ASR transcriptions as input for the NLU subcomponent which did not outperform the E2E SLU model in table IX. Analysis showed that the character-based ASR E2E approach made better use of a reduced amount of data than the pipeline word-based approach for which more data is needed. It also trains better on combined natural and artificial speech. This also demonstrates that a high ASR performance is not mandatory for an E2E SLU approach, different from the pipeline SLU.

VI. DISCUSSION

E2E SLU is only partially dependent on ASR performance, and intent prediction can benefit from a *well-balanced* attention-CTC multi-task learning (optimal results were obtained with a multi-task learning weight of *0.5*). The attention mechanism combined with the bi-LSTM allows a more flexible alignment, which focuses on the important parts (the intent label symbols) in the sequence and models long-term dependencies from which intent prediction can benefit. However erroneous ASR transcriptions have an impact on intent prediction. For E2E ASR frequent errors occur for the keyword proper noun predictions (10% of the total ASR errors), different from pipeline ASR with a lexicon. Mispronunciations in the artificial speech data partially explain these errors but have their impact on intent classification as each command contains a keyword. Hence by moving a small portion of real domain-specific data to the training data these errors decreased.

TABLE IX
PIPELINE AND E2E SLU PERFORMANCES (%) WITH VOCADOM@A4H SUBSET (1K.) IN TRAINING AND SUB-SAMPLING

Training set (reduced)	Hours of speech	(%) TTS in train	Intent F1-score
Pipeline SLU	84.69	94.39	61.35
E2E SLU	84.69	94.39	**70.21**

To reduce the impact of imbalanced data, in the pipeline approach, a weighting majority class strategy was used successfully in the cross entropy loss function. In the E2E SLU model, data over- and sub-sampling of the minority and majority classes was applied to the training data, improving performances. Although the ASR performance must be improved, it shows that E2E spoken intent recognition is feasible with imperfect ASR transcriptions, if the ratio between natural and artificial speech in a small unaligned training data set is optimal. The E2E spoken intent recognition approach did not outperform the pipeline approach. However, the best model still obtains a 70.21% F1-score for intent prediction without using the slot label information, contrary to the pipeline approach using the named entity information at the same time as the intents in a multitask setting. On top of that the pipeline approach was outperformed by the E2E SLU approach with both systems trained on the same small-sized training data (61.35% vs 70.21% F1-score).

VII. CONCLUSION AND FUTURE WORK

This study shows that E2E intent prediction is possible in a data scarce context combining NLG and TTS augmentation. Furthermore it is portable to new domains, providing there is a small amount of domain-specific data. These aspects have not been investigated in the closest related work to ours [2], [3]. E2E intent prediction is a promising way to reach similar or higher performances than a pipeline approach. Further work to achieve this includes extending our intent recognition approach with slot label and slot value information also by using transcription augmentation. On top of that, multi-task [3] and transfer learning with models trained on similar or far larger domain-specific data sets should be investigated.

VIII. ACKNOWLEDGEMENTS

We thank Dr. Raheel Qader and Dr. Benjamin Lecouteux for their support using the PyTorch seq2seq library and Kaldi.

REFERENCES

[1] Y. Wang, L. Deng, and A. Acero, "Semantic frame-based spoken language understanding," in *Spoken language understanding: systems for extracting semantic information from speech*. Wiley, 2011.

[2] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," *arXiv preprint arXiv:1802.08395*, 2018.

[3] S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin, "End-to-end named entity and semantic concept extraction from speech," in *IEEE Spoken Language Technology Workshop*, Athens, Greece, 2018.

[4] L. Zhai, P. Fung, R. Schwartz, M. Carpuat, and D. Wu, "Using n-best lists for named entity recognition from Chinese speech," in *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, 2004, pp. 37–40.

[5] K. Sudoh, H. Tsukada, and H. Isozaki, "Incorporating speech recognition confidence into discriminative named entity recognition of speech data," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 617–624.

[6] E. Simonnet, S. Ghannay, N. Camelin, Y. Estève, and R. De Mori, "ASR error management for improving spoken language understanding," in *Proceedings of Interspeech 2017*, Stockholm, Sweden, 2017.

[7] M. Hatmi, C. Jacquin, E. Morin, and S. Meigner, "Incorporating named entity recognition into the speech transcription process," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech'13)*, 2013, pp. 3732–3736.

[8] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[9] J. Li, R. Gadde, B. Ginsburg, and V. Lavrukhin, "Training neural speech recognition systems with synthetic speech augmentation," *arXiv preprint arXiv:1811.00707*, 2018.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[11] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[12] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and others, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 3, pp. 530–539, 2015.

[13] A. Bapna, G. Tur, D. Hakkani-Tur, and L. Heck, "Sequential dialogue context modeling for spoken language understanding," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017.

[14] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proceedings of Interspeech 2016*, 2016, pp. 685–689.

[15] L. Huang, A. Sil, H. Ji, and R. Florian, "Improving slot filling performance with attentive neural networks on dependency structures," *arXiv:1707.01075 [cs]*, 2017.

[16] 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.

[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 *International Conference on Text, Speech, and Dialogue*. Springer, 2018, pp. 509–517.

[18] A. Mishakova, F. Portet, T. Desot, and M. Vacher, "Learning Natural Language Understanding Systems from Unaligned Labels for Voice Command in Smart Homes," in *The 1st International Workshop on Pervasive Computing and Spoken Dialogue Systems Technology (PerDial 2019)*, Kyoto, Japan, 2019.

[19] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[20] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, vol. 5, 2015, pp. 1–6.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[22] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[23] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proceedings of the International conference on machine learning*, 2016, pp. 173–182.

[24] 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 *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC)*, 2014, pp. 4499–4506.

[25] F. Portet, S. Caffiau, F. Ringeval, M. Vacher, N. Bonnefond, S. Rossato, B. Lecouteux, and T. Desot, "Context-Aware Voice-based Interaction in Smart Home -VocADom@A4H Corpus Collection and Empirical Assessment of its Usefulness," in *17th IEEE International Conference on Pervasive Intelligence and Computing (PICom 2019)*, Fukuoka, Japan, 2019.

[26] Y. Hou, Y. Liu, W. Che, and T. Liu, "Sequence-to-sequence data augmentation for dialogue language understanding," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1234–1245.

[27] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, no. 1, 2018.

[28] N. Serpollet, G. Bergounioux, A. Chesneau, and R. Walter, "A large reference corpus for spoken French: Eslo 1 and 2 and its variations," in *Proceedings from Corpus Linguistics Conference Series, University of Birmingham*, 2007.

[29] V. Rangarajan and S. Narayanan, "Analysis of disfluent repetitions in spontaneous speech recognition," in *Signal Processing Conference, 2006 14th European*. IEEE, 2006, pp. 1–5.

[30] T.-P. Tan and L. Besacier, "A French non-native corpus for automatic speech recognition." in *Proceedings of the Language Resources and Evaluation Confere (LREC)*, vol. 6, 2006, pp. 1610–1613.

[31] F. Aman, M. Vacher, S. Rossato, R. Dugheanu, F. Portet, J. Grand, and Y. Sasa, "Etude de la performance des modèles acoustiques pour des voix de personnes âgées en vue de l'adaptation des systèmes de RAP (assessment of the acoustic models performance in the ageing voice case for ASR system adaptation)[in French]," in *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 1: JEP*. ATALA/AFCP, 2012, pp. 707–714.

[32] M. Vacher, S. Bouakaz, M.-E. Bobillier-Chaumon, F. Aman, R. A. Khan, S. Bekkadja, F. Portet, E. Guillou, S. Rossato, and B. Lecouteux, "The CIRDO corpus: comprehensive audio/video database of domestic falls of elderly people," in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. ELRA, 2016, pp. 1389–1396.

[33] G. Perennou, "B.D.L.E.X. : A data and cognition base of spoken French," in *Proceedings of ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, 1986, pp. 325–328.