SLU FOR VOICE COMMAND IN SMART HOME: COMPARISON OF PIPELINE AND END-TO-END APPROACHES

Thierry Desot, François Portet, Michel Vacher

SLU FOR VOICE COMMAND IN SMART HOME: COMPARISON OF PIPELINE AND END-TO-END APPROACHES

Thierry Desot, François Portet, Michel Vacher

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble France

ABSTRACT

Spoken Language Understanding (SLU) is typically performed through automatic speech recognition (ASR) and natural language understanding (NLU) in a pipeline. However, errors at the ASR stage have a negative impact on the NLU performance. Hence, there is a rising interest in End-to-End (E2E) SLU to jointly perform ASR and NLU. Although E2E models have shown superior performance to modular approaches in many NLP tasks, current SLU E2E models have still not definitely superseded pipeline approaches.

In this paper, we present a comparison of the pipeline and E2E approaches for the task of voice command in smart homes. Since there are no large non-English domain-specific data sets available, although needed for an E2E model, we tackle the lack of such data by combining Natural Language Generation (NLG) and text-to-speech (TTS) to generate French training data. The trained models were evaluated on voice commands acquired in a real smart home with several speakers. Results show that the E2E approach can reach performances similar to a state-of-the art pipeline SLU despite a higher WER than the pipeline approach. Furthermore, the E2E model can benefit from artificially generated data to exhibit lower Concept Error Rates than the pipeline baseline for slot recognition.

***Index Terms*—** Spoken language understanding, automatic speech recognition, natural language understanding, ambient intelligence, voice-user interface

1. INTRODUCTION

Spoken Language Understanding (SLU) systems typically consist of a pipeline of automatic speech recognition (ASR) and natural language understanding (NLU) modules. For a *slot-filling* task, ASR output transcriptions are fed to the NLU module for slot and intent extraction. The NLU model is trained on clean transcriptions whereas erroneous ASR transcriptions reduce the SLU 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 [1, 2]. The main motivation for applying E2E SLU is that word by word recognition is not necessary to infer slots and intents and that the ASR phoneme dictionary and language model (LM) become optional. E2E SLU systems are deep recurrent neural networks (RNN) that can be learned on GPUs to train directly from a large amount of speech data. Large amounts of domain-specific data are difficult to acquire, limiting the applicability of E2E to new domains.

In this paper, we compare pipeline and E2E SLU for voice command in smart homes. The E2E model is trained on transcriptions enriched with symbols representing intents and slots. We show that, contrary to the pipeline approach, intents and slots can be inferred directly from the raw speech input, using a small training data set. The contributions of this paper are: 1) the first work on E2E SLU for voice command in domestic environments; 2) a comparison of a pipeline and E2E SLU model; 3) experiments performed with realistic non-English test and synthetic training data. Both approaches are positioned with respect to the state-of-the-art in Section 2 and are outlined in Section 3. 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 3 and 4. Section 5 presents the results of experiments using real smart home voice commands for the SLU pipeline baseline and E2E approaches followed by a discussion, conclusion and outlook on future work.

2. RELATED WORK

A typical SLU pipeline approach is *sequential* and is composed of an ASR and NLU module. The ASR output hypotheses from the speech utterance are fed into the NLU model aiming to extract the meaning from the input transcription. In this paper, SLU is seen as a slot-filling task for prediction of the speaker's *intent* and entity labels (*slots*) in a spoken utterance [3]. The main issue of the state-of-the art pipeline SLU approaches is to address the cascading error effect between the ASR and the NLU module. Such sequential approaches used confidence measures and N-best lists to decrease the errors accumulated over the ASR and NLU modules. For instance, weighted voting strategies combining ASR output

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

confidence measures and N-best list hypotheses were used in a NER task [4] to take uncertainty into account. A named entity (NE) label was considered correct if it occurred in more than 30% of the n-best candidates. Another method is to learn NLU models on noisy ASR output 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) NE recognition system.

More recently, acoustic word embeddings for ASR error detection were trained through a convolutional neural network (CNN) based ASR model to detect erroneous words [6]. Output of this ASR model was fed to a conditional random fields (CRF) model and an attention-based RNN NLU model. The CRF outperformed the RNN approach and the concept error rate (CER) decreased further by integrating confidence measures. If most previous approaches for SLU focused on tuning the ASR model or using N-best hypotheses, other work [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 SLU was conceived as a *parallel* or joint processing of the ASR and NLU tasks. These E2E approaches integrated deep neural networks (DNN) inferring intents directly from audio MFCC features training a sequence-to-sequence (seq2seq) model on clean and noisy speech data [8], but did not outperform the pipeline approach. In [9] the Baidu Deep Speech ASR system [10] was trained on NE annotated transcriptions. The training set was increased by performing NER on a large speech data set and exhibited a better identification of slot labels than a pipeline system. Although E2E model performances are promising, pipeline models are still highly competitive. Furthermore training data augmentation, is a key factor for bringing E2E SLU to performances equivalent or superior to pipeline SLU performances. Generation of training data using TTS was shown to be useful for ASR [11]. Gadde *et al.* used an ASR E2E convolutional NN model with connectionist temporal classification (CTC) and reported optimal ASR performances with 50% synthetic and 50% natural speech data in the acoustic model. This review of related work shows that the E2E SLU field needs more research effort indicating that training data generation seems to be crucial. This aspect is developed in section 5.2.

3. PIPELINE AND E2E SLU METHODS

3.1. Pipeline SLU

The ASR component of our pipeline SLU is based on the hybrid HMM-DNN Kaldi tool using speaker adapted features from the Gaussian mixture model (GMM) [12]. The nnet2 version was used to benefit from multiple GPUs [13]. Its output transcriptions are fed to an NLU module.

State-of-the-art NLU CRF models [14] and also DNN-based models [15, 16, 17, 18] approach the NLU problem as a *sequence labeling task*. This means that the training data must be *aligned* to associate each word to a slot label as in the *BIO* NE labeling scheme. In spite of its efficiency in a pure NLU context, this type of alignment cannot be assumed for pipeline and E2E SLU when input data consists of spontaneous speech with disfluencies that often cause ASR deletion and insertion errors. Hence a robust NLU sub-part is needed that can handle those and is trainable with unaligned labels. Therefore we used a *sequence generation* approach to make the training become independent from aligned data. Using unaligned data provides the flexibility to infer slot labels from imperfect transcriptions. Hence, in this work, the NLU module was a seq2seq attention-based model [1].

Although our seq2seq NLU model is close to the one of Liu *et al.* [17] showing high performances on a voice command task using aligned data [19], it should learn to associate several words to one slot label without aligned data. For instance, from *"Turn on the light" (Allume la lumière)* the model generates the sequence `intent[set_device], action[turn on], device[light]`, without specifying the slot associated with the definite article. An aligned approach however predicts two labels for the sequence *"the light"*, one label for the article and another for the noun as parts of the same named entity. Furthermore, *perfect* ASR is not necessary for training the NLU model. In the (erroneous) ASR hypothesis transcription *"turn on light"* for the reference transcription *"turn on the light"*, the information about the slots `action` and `device` , is still available.

3.2. E2E SLU

The E2E approach is based on ESPnet [20]. It integrates the KALDI data preparation, extracts Mel filter-bank features, and combines Chainer and PyTorch deep learning tools [21, 22]. The default PyTorch encoder is a pyramidal subsampling bi-LSTM (BLSTM) [23] given T-length speech feature sequence $o_{1:T}$ to extract feature sequence $h_{1:T'}$ as,

$$h_{1:T'} = BLSTM(o_{1:T}), \tag{1}$$

where $T' < T$ due to subsampling. The chainer back-end supports CNNs. Mapping from acoustic features to character sequences is performed by a *hybrid* multitask learning that combines CTC [24] and an attention-based encoder-decoder. The attention mechanism allows a more flexible alignment, which focuses on the important features and character sequences whereas the ASR alignment is monotonic. A trade-off hybrid CTC and attention-based approach finds a balance between attention and CTC.

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

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

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. 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:

$$\begin{aligned} logp(y_n|y_{1:n-1}, h_{1:T'}) = logp^{hyb}(y_n|y_{1:n-1}, h_{1:T'}) \\ + \beta\, logp^{lm}(y_n|y_{1:n-1}). \end{aligned} \quad (3)$$

Since ESPnet models the ASR task at the character level, our approach to predict NLU concepts from the input signal was inspired by [9]. The output target of the ESPnet process was speech transcriptions augmented with characters (e.g., `@`, `#...`) symbolizing the intent and slot labels of the utterance, described in section 5.2.

4. DATA

In spite of the development of voice based IoT devices, there is a lack of domain-specific speech data, especially for non-English languages. This problem was tackled with the automatic generation of a domain-specific *synthetic speech training* corpus. To evaluate the method, two French voice command data sets were collected in real smart homes (section 4.2) and are available to the community [25, 26].

4.1. Data augmentation using artificial data generation

Although the current trend for data augmentation is to use constrained RNN language models [27], these still need a set of initial sentences for bootstrapping and it is difficult to make them generalize to unseen concepts. For that reason we used standard expert-based NLG [28]. The corpus generator of Desot *et al.* [19] produced training data automatically labeled with intents and slots. 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.

Its semantics were developed around an existing smart home Amiqual4Home[2]. Intents consist of four general categories: `contact` which allows the user to place a call; `set` to change the state of objects in the smart home; `get` to query the state of objects and properties of the world at large (open question); and `check` to check the state of an object (closed question). A complete overview of intents is presented in table 1.

Eight different basic slot labels were defined: 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. Variations on these basic slot labels occur. Each generated voice command is composed of a keyword 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"* will activate the Smart Home whereas *"Call a doctor"* should not trigger any reaction. More than 77k semantically annotated voice commands were automatically produced for training purpose, with a total of 17 different slot labels and 7 different intent classes.

4.2. Collection of realistic data sets

Evaluation of voice command SLU requires real data. Hence, we used the VocADom@A4H corpus [26] with about twelve hours of audio signal[3]. It was acquired in realistic conditions in the Amiqual4Home smart home. More than 150 sensors and actuators were set to acquire speech, to control light, set the heating etc. Eleven participants uttered voice commands while performing daily activities 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. Speech data was semi-automatically transcribed and resulted in 6,747 utterances, annotated with intents and slots. The small SWEET-HOME corpus, with distant voice commands collected in another smart home [25] was also used.

Finally, we used ESLO2 corpus utterances (126h) of conversational French speech [29] to model the `none` intent. Similar to VocADom@A4H and SWEET-HOME corpora, it contains frequent disfluencies. From the ESLO2 data, sentences which were unrelated to voice command intent were extracted (i.e. `None` intent). By using an n-gram model learned on the artificial corpus, sentences with domain specific vocabulary were selected. These utterances were manually filtered and only out of domain utterances were kept in order to collect `none` intent training data. Table 2 summarizes the statistics for the artificial (Artif.), VocADom@A4H (VocADom.), Sweet-Home and Eslo2 corpora. The VocADom@A4H corpus was used as *test* corpus in all the experiments (unless otherwise specified).

5. EXPERIMENTS AND RESULTS

5.1. Pipeline SLU baseline approach

Baseline ASR transcriptions were generated using Kaldi. We wanted to evaluate the impact of conversational speech in the training data, similar to the conversational style of the VocADom@A4H test data. That is why two acoustic models were compared: a first model without and a second model with ESLO2 speech data. The first one was trained on 90% of the corpora ESTER1 (100h) and 2 (100h), REPERE (60h),

[2]https://amiqual4home.inria.fr

[3]https://vocadom.imag.fr

Table 1. Artificial corpus (Artif.) and VocADom@A4H (Real.): Examples and Frequency of intents

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

Table 2. Comparison of the corpora used for SLU

Parameters	**Artif.**	**VocADom.**	**Sweet-Home**	**Eslo2**
utterances	77,481	6747	1412	161,699
words	187	1462	480	29,149
intents	7	8	6	1
slot labels	17	14	7	-

ETAPE (30h), SWEET-HOME (2.5h), BREF120 (120h) [30], VOIX-DETRESSE (0.5h) [31] and CIRDOSET (2h) [32], the remaining 10% being kept as development (DEV) set. For the second one, we added 90% of the speakers of 126 hours of ESLO2 speech data [29] to the first training corpus, 10% being added to the first DEV set. The division into train and DEV set was based on random speaker selection without speaker overlap between the training corpus and 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* [34]. For decoding, we used a domain-specific 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 and Wikipedia. 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 that 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 3 show that the fMLLR and HMM-DNN models with the ESLO2 data outperform the acoustic models without it. The WER is slightly superior than a recent study using a similar approach for French voice command recognition in a smart home [35].

The NLU seq2seq model was a bi-directional LSTM encoder and decoder. 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 training data was 90% of the combined semantically annotated data sets : artificial, SWEET-HOME and the filtered ESLO2 utterances without intent; the remaining 10% being the DEV set. The test data was the VocADom@A4H corpus. Both sets are described in section 4. Similar to [6] we report NLU performances on the VocADom@A4H test set using the concept error rate (CER) for slot labels. Intent classification is evaluated using the F1-score. As the NLU problem is designed as a sequence generation task using unaligned data, the type of errors differs from a sequence labeling task with aligned data as used in [19]. Typical errors using aligned data are substitutions whereas with unaligned data, frequent deletions and insertions occur. In [36] the CER is defined as the ratio of the sum of deleted, inserted and confused concepts w.r.t. a Levenshtein-alignment for a given reference concept string. We calculated the CER in a similar way, but we did not take the label sequence order into account since a reference sequence `action, device` provides the same information as a hypothesis `device, action`.

Evaluation metrics for intent and slot label level on VocADom@A4H are shown in table 4. Results analysis shows a strong tendency towards `none` intent predictions due to the majority `none` intent class (*unweighted manual*). A modification of the weight assignment in the cross entropy loss func-

Table 3. ASR performance (WER %) on test set

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

tion of the NLU model handled this imbalanced data problem. 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 (4).

$$weight\_class\_i = \frac{total\_instances}{instances\_class\_i} \quad (4)$$

This method improved performances (*weighted manual*). NLU performances for intent prediction on the VocADom@A4H ASR output (*weighted ASR*) are slightly worse as compared to the manual transcription predictions (*weighted manual*), and significantly worse for slot predictions.

5.2. SLU E2E approach

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. This section reports the performance of ESPnet on an ASR task followed by an SLU task.

In [11] an E2E deep convolutional NN model using CTC was trained for an ASR task, on a natural speech data set augmented with synthetic speech. The synthetic data combined with the natural speech data with an optimal ratio has proven to be beneficial to the ASR performance. To compensate the lack of a large amount of domain-specific speech training data, we applied a similar technique. The speech data set used for the ASR module of the pipeline SLU approach (section 5.1) is also the training data for the E2E SLU approach (*real_data* in Table 5) but is augmented with TTS data. To that end synthetic speech was generated on the complete artificial corpus using the open source French female SVOX voice and represents 14.67% of the resulting total acoustic model speech data[4]. As a preparatory phase for the SLU task, the ASR model was trained to evaluate the impact of the synthetic speech in the training data and to estimate the DNN parameters.

Table 5 reports the results on the VocADom@A4H test set. The first raw of the table (*real_data*) exhibits a far worse WER as compared to the baseline Kaldi ASR model (trained on real speech data only). However, decoding using a character-based LM created with the same data as used for the pipeline ASR LM (section 5.1) improved the WER (*real_data+LM*) while the addition of the TTS generated data (*real_data+LM+TTS*) provided another significant improvement. Although far from perfect, the results obtained on our DEV set (25.7% WER) are comparable to those obtained by Ghannay *et al.* [9] on their DEV set (20.70% WER) using the Baidu Deep Speech E2E ASR system.

To perform E2E SLU, we injected the intent and slot label symbols (section 4.1) into the clean transcriptions of the 524k (553.9 hours of speech) training data utterances as used for the E2E ASR. In this way the 14% of artificial data and 7% of the real speech data (without intent) sentences were enriched with slot and intent symbols. The symbolic slot label annotations were bootstrapped from the artificial data to the utterances in the training data without intent. In the sentence *"The light is switched off" (La lumière est éteinte)*, the slot label for *"The light"* is `device`. For `none` intent sentences without voice command, no symbolic intent annotation was inserted. Due to space limitation, we do not present the complete overview of symbols. Table 6 provides an example of the voice command *"VocADom switch on the light"*, symbolically annotated with intent and slot labels.

To study the impact of the synthetic data on the SLU experiments, different proportions of TTS generated speech were used in the training data for the VocADom@A4H test set. The character-based LM was generated using the training data with the intent and slot labels injected as symbols. The first row in Table 7 recalls the best baseline pipeline performance from Table 4. The second row shows E2E SLU performance with a model trained on the artificial speech data (*artificial only*) composed of only 81.25 hours of speech. The last row presents the results using the complete training data (*complete*) with the character based LM for decoding. These results show that the pipeline approach is by far superior to the E2E approach. However, this difference is partially due to the `none` intent class being over-represented in the *complete* data set. Imbalanced data for the pipeline SLU was dealt with using a weighted learning scheme of the pipeline. In the E2E case we handled this with sub-sampling. The `none` intent class instances were decreased leaving only 11k utterances with a `none` class label for training. On top of that 1k sentences were moved from the test set to the training set to evaluate the impact of increased real domain-specific data in the speech base. This resulted in a data set of 84.69 hours of

Table 4. Pipeline performances (%) on VocADom@A4H

Model	**Intent F1-score**	**Slot CER**
unweighted manual	76.95	42.67
weighted manual	**85.51**	**33.78**
weighted ASR	**84.21**	**36.24**

Table 5. ESPnet ASR performances on VocADom@A4H

Model	**WER** (%)
real_data	53.5
real_data+LM	50.6
real_data+LM+TTS	**46.5**

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

Table 6. Intent and slot label symbols for E2E SLU

Intent
@ VocADom switch on the light @ `set_device`
Slot labels
VocADom ˆswitch onˆ }the light} `action device`

speech with a portion of 94.39% artificial data.

For a fair comparison the pipeline SLU ASR and NLU modules were retrained with the same reduced data. Performance of the pipeline SLU *baseline* and *E2E SLU* on this small training data set with a small portion of domain-specific real data is reported in Table 8. It shows significantly improved E2E intent and slot label prediction performances. On top of that it supersedes the baseline pipeline approach. The maximal E2E SLU performance was reached using an attention-CTC multi-task learning weight of 0.5. For the pipeline ASR (Kaldi) a WER of >90% was exhibited, showing a too large distance between test and artificial training data acoustic features. However, with an E2E ASR (ESPnet) training on the same reduced data a WER of 60.6% was obtained. With the ESPnet ASR transcriptions as input the NLU subcomponent did not outperform the E2E SLU (*baseline* and *E2E SLU* in Table 8). This shows that the E2E approach made better use of a reduced amount of data and that high ASR performance is not mandatory for an E2E SLU approach.

Table 7. E2E SLU performances (%) on VocADom@A4H

Train set	**Hours of speech**	**(%) TTS in train**	**Intent F1-score**	**Slot CER**
baseline	472.65	0	**84.21**	**36.24**
artifical only	81.25	100	35.94	56
complete	553.9	14.67	47.31	51.87

Table 8. E2E SLU performances (%) with VocADom@A4H subset (1k.) in training and sub-sampling

Train set	**Hours of speech**	**(%) TTS in train**	**Intent F1-score**	**Slot CER**
baseline	84.69	94.39	61.35	35.62
E2E SLU	84.69	94.39	**70.21**	**26.17**

6. DISCUSSION

E2E SLU is only partially dependent on ASR performance. Pearson's correlation coefficient between the E2E ASR model WER in Table 5 (*real_data+LM+TTS*) and the E2E SLU model CER in Table 7 (*complete*) on the test set shows a low correlation (r = 0.33). This is confirmed by significantly higher E2E SLU than E2E ASR performances on the reduced small data set (*E2E SLU* in Table 8). Intent and slot label prediction benefits from a well-balanced attention-CTC multi-task learning. The desired ASR alignment is monotonic, but less needed for slot and especially for intent prediction. The attention mechanism combined with the bi-LSTM allows a more flexible alignment, which focuses on the important parts (the slot and intent label symbols) in the sequence and models long-term dependencies, necessary for intent prediction. However morphological ASR errors such as the imperative mood, e.g *"turn off" (éteins)*, being substituted by the indicative mood, *"turns off" (éteint)* decreased E2E SLU performances. One third of the intents in the test set have an imperative mood verb. Different from the pipeline ASR performance with a lexicon, frequent E2E ASR (without lexicon) errors occur for the keyword proper noun predictions (10% of the total ASR errors). These are partially due to mispronunciations in the artificial speech data. Moving a small portion of real domain-specific data to the training data reduced these types of errors. Imbalanced data was handled with a weighting majority class strategy in the cross entropy loss function of the pipeline SLU NLU module. In the E2E SLU model, data sub-sampling of the majority classes was applied to the training data. The improved E2E SLU performances on the resulting reduced data (Table 8) as compared to the lower E2E ASR performances (60.6% WER) demonstrate the feasibility of E2E SLU with far from perfect ASR transcriptions. This is possible with an optimal ratio between natural and artificial speech in a small unaligned training data set.

7. CONCLUSION AND FUTURE WORK

Our E2E SLU best model obtains a 70.21% F1-score for intent prediction, and outperforms the SLU pipeline approach for slot prediction with a CER of 26.17% using a small training data set. This study shows that E2E SLU is feasible with scarce domain-specific data, portable to new domains, combining NLG and TTS augmentation with far from perfect ASR. These aspects have not been investigated in the closest related work to ours [8, 9]. E2E SLU is a promising way to reach equivalent or higher performances than a pipeline approach. Further work to achieve this, includes multi-task [9], transfer and curriculum learning with models trained on similar or larger domain-specific data sets.

Acknowledgement

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

8. REFERENCES

[1] Swapnil Bhosale, Imran Sheikh, Sri Harsha Dumpala, and Sunil Kumar Kopparapu, "End-to-end spoken language understanding: Bootstrapping in low resource scenarios," *Proc. Interspeech 2019*, pp. 1188–1192, 2019.

[2] Natalia Tomashenko, Antoine Caubriere, and Yannick Esteve, "Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech," *Interspeech, Graz, Austria*, 2019.

[3] Ye-Yi Wang, Li Deng, and Alex Acero, "Semantic frame-based spoken language understanding," in *Spoken language understanding: systems for extracting semantic information from speech*. Wiley, 2011.

[4] Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai 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] Katsuhito Sudoh, Hajime Tsukada, and Hideki 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] Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève, and Renato De Mori, "ASR error management for improving spoken language understanding," in *Proceedings of Interspeech 2017*, Stockholm, Sweden, 2017.

[7] Mohamed Hatmi, Christine Jacquin, Emmanuel Morin, and Sylvain 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] Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio, "Towards end-to-end spoken language understanding," *arXiv preprint arXiv:1802.08395*, 2018.

[9] Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, and Emmanuel Morin, "End-to-end named entity and semantic concept extraction from speech," in *IEEE Spoken Language Technology Workshop*, Athens, Greece, 2018.

[10] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[11] Jason Li, Ravi Gadde, Boris Ginsburg, and Vitaly Lavrukhin, "Training neural speech recognition systems with synthetic speech augmentation," *arXiv preprint arXiv:1811.00707*, 2018.

[12] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," Tech. Rep., IEEE Signal Processing Society, 2011.

[13] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[14] 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.

[15] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong 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.

[16] Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck, "Sequential dialogue context modeling for spoken language understanding," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017.

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

[18] Lifu Huang, Avirup Sil, Heng Ji, and Radu Florian, "Improving slot filling performance with attentive neural networks on dependency structures," *arXiv:1707.01075 [cs]*, 2017.

[19] Thierry Desot, Stefania Raimondo, Anastasia Mishakova, François Portet, and Michel 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.

[20] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "Espnet: End-to-end speech processing toolkit," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2207–2211, 2018.

[21] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin 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)*, 2015, vol. 5, pp. 1–6.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[23] William Chan, Navdeep Jaitly, Quoc Le, and Oriol 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.

[24] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang 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.

[25] Michel Vacher, Benjamin Lecouteux, Pedro Chahuara, François Portet, Brigitte Meillon, and Nicolas 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.

[26] François Portet, Sybille Caffiau, Fabien Ringeval, Michel Vacher, Nicolas Bonnefond, Solange Rossato, Benjamin Lecouteux, and Thierry 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.

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

[28] Albert Gatt and Emiel 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.

[29] Noëlle Serpollet, Gabriel Bergounioux, Annie Chesneau, and Richard 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.

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

[31] Frédéric Aman, Michel Vacher, Solange Rossato, Remus Dugheanu, François Portet, Juline Grand, and Yuko 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] Michel Vacher, Saida Bouakaz, Marc-Eric Bobillier-Chaumon, Frédéric Aman, Rizwan Ahmed Khan, S Bekkadja, François Portet, Erwan Guillou, Solange Rossato, and Benjamin 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*, 1986, vol. 11, pp. 325–328.

[34] Frederic Bechet, "LIA PHON : un système complet de phonétisation de textes," *Traitement Automatique des Langues (TAL)*, vol. 42, no. 1, pp. 47–67, 2001.

[35] Benjamin Lecouteux, Michel Vacher, and François Portet, "Distant Speech Processing for Smart Home Comparison of ASR approaches in distributed microphone network for voice command," *International Journal of Speech Technology*, vol. 21, pp. 601–618, 2018.

[36] Stefan Hahn, Patrick Lehnen, Christian Raymond, and Hermann Ney, "A comparison of various methods for concept tagging for spoken language understanding," in *LREC 2008*, 2008.