A hybrid convolutional and recurrent network approach for conversational AI in spoken language understanding

1st Bassel Zaity
Graduate School of Software Engineering
Peter the Great St.Petersburg Polytechnic University (SPbPU)
Saint Petersburg, Russia
bassel.zaity@gmail.com

2nd Hazem Wannous
IMT Lille Douai
CRIStAL UMR 9189, University of Lille
Lille, France
hazem.wannous@univ-lille.fr

3rd Zein Shaheen
Computer Intelligent Technologies
Peter the Great St.Petersburg Polytechnic University (SPbPU)
Saint Petersburg, Russia
shahin.z@edu.spbstu.ru

4th Igor Chernoruckiy
Graduate School of Software Engineering
Peter the Great St.Petersburg Polytechnic University (SPbPU)
Saint Petersburg, Russia
igcher@spbstu.ru

5thPavel Drobintsev
Graduate School of Software Engineering
Peter the Great St.Petersburg Polytechnic University (SPbPU)
Saint Petersburg, Russia
drob@ics2.ecd.spbstu.ru

6th Vadim Pak
Computer Intelligent Technologies
Peter the Great St.Petersburg Polytechnic University (SPbPU)
Saint Petersburg, Russia
vadim.pak@cit.icc.spbstu.ru

***Abstract*—The deep learning revolution has an impact on almost all parts of our life, it brought us improved momental machine translators, modern human-like conversation voice assistant like Siri, Alexa, Alisa. This revolution had become truth because of deep learning methods which improved multiple processing layers to learn a hierarchical representation of data, and have achieved the state-of-the-art results in many lives domains. In this paper, we are focusing on one of the most famous NLP (Natural language processing) problems which is slot filling to approach the state-of-the-art results on the ticketing problem to make the Spoken Dialogue systems work more efficiently. We propose a hybrid architecture, as a combination of a Recurrent Neural Network and a Convolutional Neural Network models, for Slot Filling in Spoken Language Understanding. In particular, our network model is built from stacked units of 1-dimensional CNN (Convolutional Neural Network) across the temporal domain, which are used to train an RNN (Recurrent neural network) layer to model dependencies in the temporal domain. Experimental tests show extensive comparisons between different models for NER (Named Entities Recognition). Results demonstrate the effectiveness of hybrid models that combine benefits from both RNN and CNN architecture compared over distinct RNN and CNN models and also compared with other traditional models. Experimental results show that our model achieves F1-score of 95.11 on benchmark ATIS dataset.**

***Index Terms*—SLU, slot-filling, Hybrid CNN and RNN, Deep learning**

I. Introduction

The methodological revolution in spoken language research had been started about 20 years ago when the machine learning algorithms started to take place in the programmer society. However, the last five years brought the real change after the new deep learning architectures, which leds to a new level of solutions and the Spoken Dialogue Systems (SDS) is one of the fields which had really improved recently. SDS and chatbots are taking a wider place day by day in the scientific conferences as a case study. They already have great commercial potential according to the changing of the way humans interact with machines. The improvement of deep learning in general, and the Natural Language Processing (NLP) researchers in special, led to place a lot of difficult problems under the microscope, and the research teams over the world trying to test different architecture models to get the state-of-the-art results to solve these problems. In our days, the importance of chatbots has increased, most websites tend to have their own chatbots to communicate with customers and facilitate their work. The goal of such bots is to know users needs and give responses in their natural language. This will lead to a better understanding of the users queries when communicating with the users in a natural way throw these chatbots. It will also help to ask the users about whatever missing points they have to bring the best accurate answers, such assistants could help disabled people and bring more solutions to the market to build a more intelligent world.

The implementation of a voice assistant comes with different parts, as speech to text and text to speech models, but the most challenging part comes in the task of NLP to extract the needs of the user and to know his intent from the conversation. The processing pipeline comes here into two parts, intent classification and slot filling after the intent is known. At this stage, the bot needs to generate a response to the user and give feedback about whatever missing data there are. The whole system that organizes this process is the dialogue manager which processes the users input, extract the meaning and generates the desired response. From a research perspective, the design of spoken dialogue systems provides a number of significant challenges, as these systems depend on solving several difficult NLP and decision making tasks, and combining these into a functional dialogue system pipeline [1]. Intent detection and slot filling are usually processed separately. Intent detection can be treated as a semantic utterance classification problem, and popular classifiers like support vector machines (SVMs) [2] and deep neural network methods [3] can be applied. Slot filling can be treated as a sequence labeling task. Popular approaches to solving sequence labeling problems include maximum entropy Markov models (MEMMs) [4], conditional random fields (CRFs) [5], and recurrent neural networks (RNNs) [6] [7] [8]. Joint model for intent detection and slot filling has also been proposed in literature [9] [10]. Such joint model simplifies the spoken language understanding (SLU) system, as only one model needs to be trained and fine-tuned for the two tasks.

This work focuses on the slot-filling part by building a model that extracts information from text in a reliable way. Before the era of deep learning the task of Named Entity Recognition (NER) was solved using grammars-based models and rule-based approaches, these models have proven to achieve good results in terms of precision but fail to capture all human-text varieties and thus the recall will be bad. Probabilistic approaches came with models built on HMM, which were state-of-art for many years and achieved an impressive achievement. With the recent revolution, many deep learning methods has replaced traditional previous ones and pushed state-of-art for these tasks in. Recurrent Neural Networks (RNN) models have replaced models based on HMM, that is RNN achieved the same task in a simpler way and deep RNNs are able to capture complex representations for the input. The problem with such models was that they need to handle the input token-by-token in sequence. Therefore, such structures could not be parallelized and the models will be slow to train and inference if the neural network structure is deep. Convolution Neural Networks (CNN) added a way to extract relations between tokens by mixing them in a way similar to extracting n-grams in the traditional NLP tasks. Such architectures that contain CNN could be optimized by parallelization so adding a convolutional layer could reduce the complexity and control the size of the neural network. In this paper, we discuss different approaches to solve slot filling for ticketing task as a NER problem, and showed different architectures that contain distinct RNN, CNN or hybrid architectures ones. We conducted many experiments with different values of the hyper-parameters and different optimization methods.

II. Related work

Rule-based approaches are done manually, at first you all needed roles should be written need to achieve the goal, this operation is time-consuming and therefore not so efficient, it will be notable that the recall is not very nice because its so difficult to write all the varieties, but the positives of ruled-based approaches the precision will be quite high [11]. The most widely used formal system for modeling constituent structure in English and other natural languages is the Context-Free Grammar or CFG. A context-free grammar consists of a set of rules or productions, each of which expresses the ways that symbols of the language can be grouped and ordered together, and a lexicon of words and symbols [12].

In machine learning methods, we need a dataset of text with markup, in this dataset, each word should be assigned to a tag, this problem is known as slots filling problem. The first which we should do is making some Feature engineering, for example, see whether the word is capitalized or it is a name of a city, some cities consists of two words, maybe you check the previous or the next words (context). Probabilistic modeling and Conditional Random Field not only assume that features are dependent on each other but also considers the future observations while learning a pattern [21]. This combines the best of both HMM and MEMM. In terms of performance, it is considered previously to be the best method for entity recognition problem. Another paper studied the comprehensive investigations of RNNs for the task of slot filling in SLU. They implemented and compared several RNN architectures, including the Elman-type and Jordan-type networks with their variants [18]

III. Deep Learning methods

A. Recurrent Neural Network RNN

Recurrent Neural Networks "Fig. 1" are used for sequence modeling, it accepts input x_t at time step t and a hidden state h_t and use this hidden state to produce output y_t, and this hidden state will be passed to the next time step. So, we can think of the hidden state as a summary of the previous inputs to the neural network, we use activation function such as $tanh$ or ReLU to calculate hidden state. Output y_t is the prediction of the next tag, it would be a vector of probabilities across our vocabulary, the following formulas explain the general form of RNN:

$$h_t = f(Ux_t + Wh_{t-1}) \tag{1}$$

$$y_t = \text{softmax}(Vh_t) \tag{2}$$

Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) are used as RNN units, these units can capture long term dependency. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation [19].

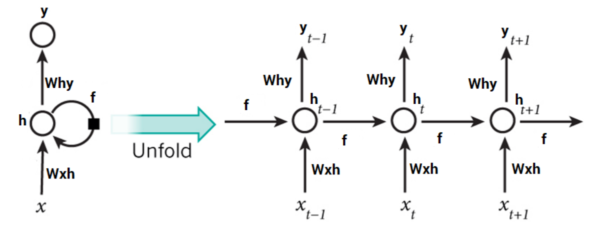

Fig. 1. General form of Recurrent Neural Network

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means let nothing through, while a value of one means let everything through! An LSTM has three of these gates "Fig. 2", to protect and control the cell state. The following formulas explain how does LSTM cell work:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4}$$

$$C_t = tanh(W_c[h_{t-1}, xt] + b_c) \tag{5}$$

$$C_t = f_t.C_{t-1} + i_t.C_t \tag{6}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t.tanh(C_t) \tag{8}$$

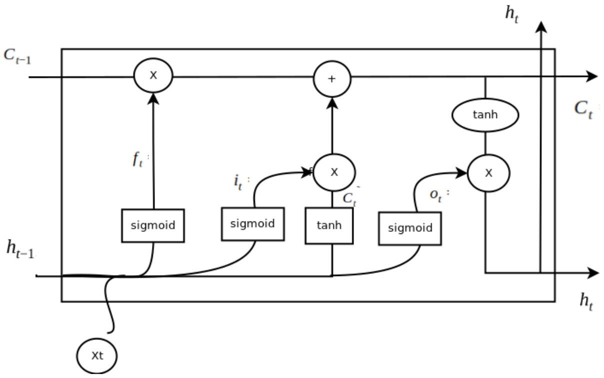

Fig. 2. LSTM unit

GRU has a simpler design "Fig. 3" it was introduced by Cho, et al. (2014) [20], The key difference between a GRU and an LSTM is that a GRU has two gates (reset and update gates) whereas an LSTM has three gates (namely input, output and forget gates) [13]. The GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit. It just exposes the full hidden content without any control. GRU is relatively new but computationally more efficient. The following formulas describe the GRU mechanism:

$$z_t = \sigma(Wz.[h_{t-1}, x_t]) \tag{9}$$

$$r_t = \sigma(Wr.[h_{t-1}, x_t]) \tag{10}$$

$$h_t = \tanh(W.[r.h_{t-1}, x_t]) \tag{11}$$

$$h_t t = (1 - z_t).h_{t-1} + z_t * h_t \tag{12}$$

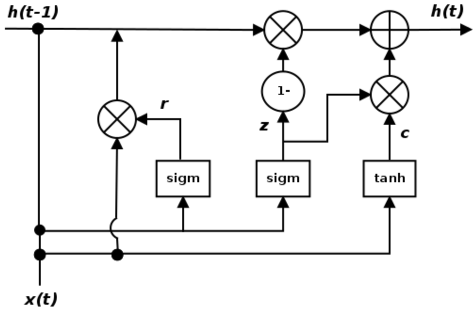

Fig. 3. GRU unit

In our experiments, we used both GRU and LSTM units and compared between them. Other sequence architectures like Encoder-decoder architecture could be used to solve this task, at first the whole input will be encoded into hidden representation (encoder), and then this hidden representation is used to produce sequence of tags (decode). Some architectures use attention mechanism to give attention to parts of the input sequence and use these information to produce the output token.

B. Convolution Neural Network for sequences

RNNs operate sequentially, the output for the second input depends on the first one and so we cant parallelize an RNN. Convolutions have no such problem, each patch a convolutional kernel operates on is independent of the other, meaning that we can go over the entire input layer concurrently. Convolutions grow a larger receptive beld as we stack more and more layers. That means that by default, each step in the convolutions representation views all of the input in its receptive field, from before and after it "Fig. 4". In our experiments we used 1D convolution to mix the tokens and extract relations between the consequence tokens, it is equivalent to n-gram relation where n is the size of the used filter, for example: if we care about the last 3 tokens we use filter size 3. Using CNN will result in some benefits, it runs faster than RNN and beats RNN in some tasks. If we divide convolution output into two parts, A and B, one of which will gate the other through element-wise multiplication, where A is liner and B through sigmoid, we get GLU (gated linear unit). Here we increased receptive field as it is shown in the following formula:

$$A = (X.W + b) \tag{13}$$

$$B = \sigma(X.V + c) \tag{14}$$

$$h_t(x) = A \oplus B \tag{15}$$

$$h_t(x) = (X.W + b) \oplus \sigma(X.V + c) \tag{16}$$

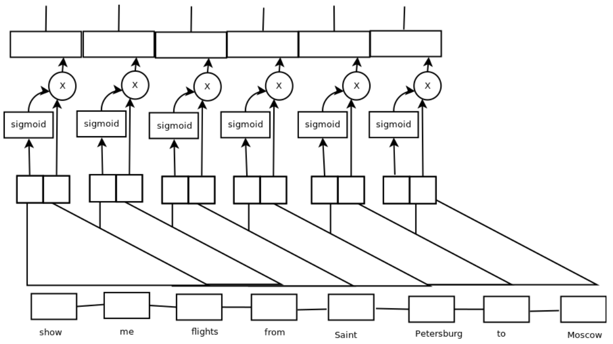

Fig. 4. Convolution Neural Network for sequences

C. Hybrid model CNN RNN

This model combines the benefits of both CNN and RNN, where RNN helps to capture the dependencies between tokens in the users query, using LSTM or GRU units will have resulted in a model that captures long-range dependencies between tokens using memory cell in their architecture. CNN will help with mixing the consequence tokens and extract relations between them [14]. In the task of slot filling, the hybrid architecture contains several convolution layers stacked with the same padding and the output of these layers will be the input for RNN layers as in Fig. 5, we can also stack several RNN layers. After these RNN layers, there will be a dense layer with softmax activations, this layer represents the output of the network.

IV. Experiments

A. Dataset

ATIS (Airline Travel Information System) corpus (Tur et al., 2010) is one of the main data resources used in many studies over the past two decades for SLU research in spoken dialog systems e.g. [15] [16] [17]. Two primary tasks in SLU are intent determination (ID) and slot filling (SF). The dataset contains audio recordings of people making flight reservations. The training set contains 4,478 utterances and the test set contains 893 utterances. We use another 500 utterances for development set. There are 120 slot labels and 21 intent types in the training set [22].

The IOB format (inside, outside, beginning) is a common tagging format for tagging tokens in a chunking task in computational linguistics, The B- prefix before a tag indicates that the tag is the beginning of a chunk, and an I- prefix before a tag indicates that the tag is inside a chunk. The B- tag is used only when a tag is followed by a tag of the same type without O tokens between them. An O tag indicates that a token belongs to no chunk.

The Table I shows an example in the ATIS dataset , with the annotation of slot/concept, named entity, intent as well as domain. The latter two annotations are for the other two tasks in SLU: domain detection and intent determination. We can see that the slot filling is quite similar to the NER task, following the IOB tagging representation, except for a more specific granularity.

TABLE I
ATIS dataset Example

Sentence	show	me	flights	from	Moscow	to	London	Today
Slots/Concepts	O	O	O	O	B-fromLoc	O	I-toLoc	B-departDate
Named Entity	O	O	O	O	S-city	O	I-city	O
Intent	Find Flight							
Domain	Airline Travel							

1) Training Details: In training, we compared between different models for NER (Named Entities Recognition) system, all the models were trained using 100 epochs. We tuned our models using different dropout values (0.1, 0.25, 0.5) and we used different optimization methods (ADAM, RMSProb, SGD). For the embedding layer, we represent each token by a vector of size 100, and for our choice for the convolution layer we used 64 filters of size 5 and used ReLU as an activation function. The hidden size of the GRU/LSTM unit is 100 “Fig. 5”.

Our architecture will go as following, input layer which is a sequence of tokens represented by indices using bag of words, embedding layer will represent each token with a vector, the vector size is a hyperparameter for the network, this embedding layer is followed by one of the main choices of the layers discussed above, recurrent neural network, convolutional neural network or a hybrid model which contains layer of CNN followed by layer of RNN.

2) Evaluation Metrics: For evaluation, we computed precision, recall and F1 score for training and validation sets, and we picked the model with the best value of the F1 score.

For Slot filling, the error rate can be computed in two ways: The more common metric is the F-measure using the slots as units. This metric is similar to what is being used for other sequence classification tasks in the natural language processing community, such as parsing and named entity extraction. In this technique, usually the IOB schema is adopted, where each of the words is tagged with their position in the slot: beginning (B), in (I) or other (O). Then, recall and precision values are computed for each of the slots. A slot is considered to be correct if its range and type are correct. The F-Measure is defined as the harmonic mean of recall and precision:

$$\text{F1-Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{17}$$

where:

$$\text{Recall} = \frac{\#\text{correct slots Found}}{\#\text{true slots}} \tag{18}$$

$$\text{Precision} = \frac{\#\text{correct slots Found}}{\#\text{found slots}} \tag{19}$$

B. Results

During evaluation process we focused on the difference between the use of different architectures of neural networks, we compared also between different optimization methods for the best neural network structure and at the end we included

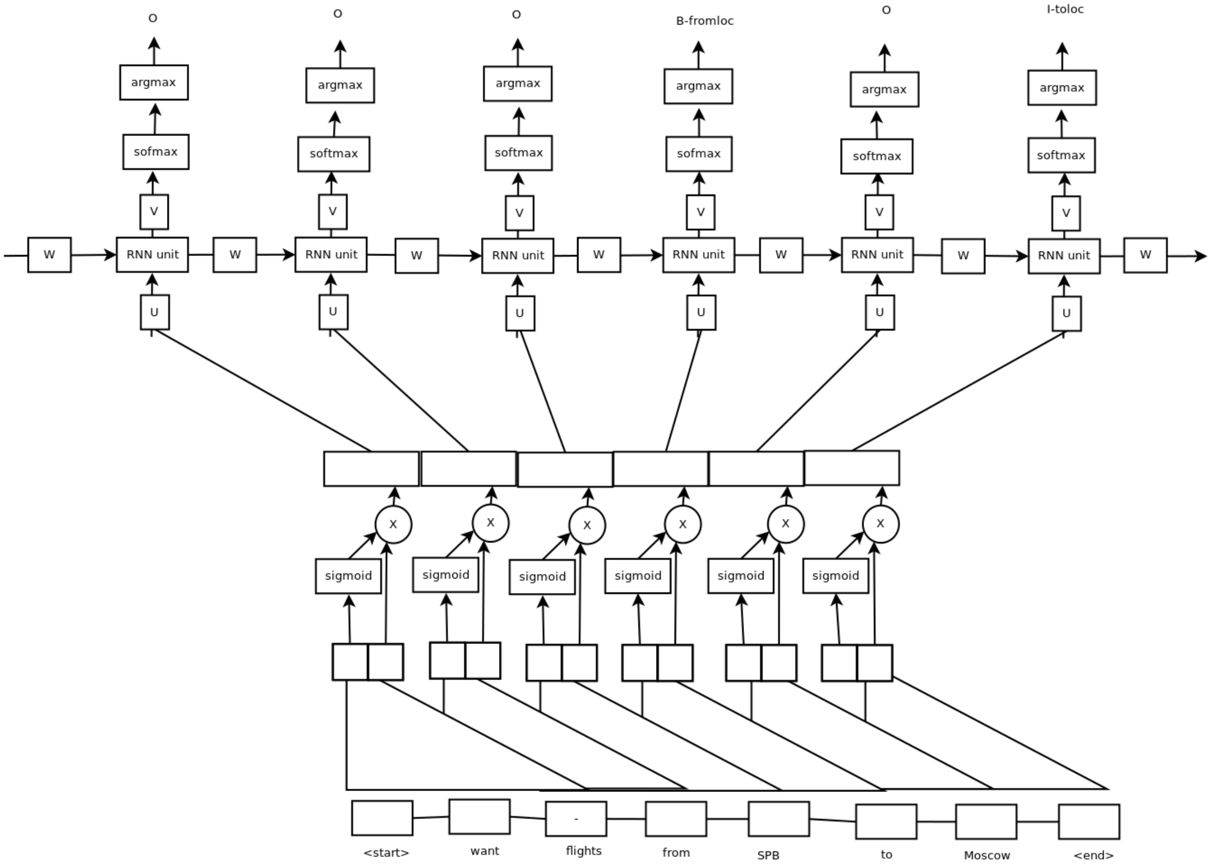

Fig. 5. Hybrid model CNN/RNN

TABLE II
COMPARISON BETWEEN DIFFERENT DEEP LEARNING STRUCTURES

Structure description	**Precision**	**Recall**	**F1-score**	**Average**
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25	94.47	95.61	95.04	94.89 ±0.15
Convolution1D structure with dropout 0.25	91.75	90.57	91.16	91.01 ±0.1
RNN/GRU structure with dropout 0.25	93.02	93.12	93.07	92.48 ±0.42

TABLE III
COMPARISON BETWEEN HYBRID STRUCTURES BASED ON OPTIMIZATION MODEL

Structure description	**Precision**	**Recall**	**F1-score**	**Average**
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25; RMSProp	94.47	95.61	95.04	94.89 ±0.15
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25; ADAM	94.71	94.95	94.83	94.67 ±0.23
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25; SGD	94.22	94.65	94.44	94.23 ±0.16

a comparison based on the type of recurrent unit used in the model. We concluded the experiments 25 times, and we took the mean of the samples and calculated the standard error. We reported our results in the tables.

Our results show that hybrid architectures perform better than other pure RNN or pure CNN models Table II, when we used dropout 0.25 and RMSProb optimization method , we got F1-score 95.04 for hybrid model compared with 91.16 for convolution model and 93.07 for recurrent model.

Our results show also that the use of RMSProb resulted in the best models according to F1-score metrics Table III, under the same dropout 0.25 and hybrid model, we got F1-score equals to 95.04 for RMSProb compared with 94.83 when we used ADAM optimization model, and 94.44 when we used SGD. Result show that the effect the Hybrid structure Convolution1D and RNN/GRU without dropout using RMSProp optimizer is giving the best F1-score 95.11 comparing with different levels of dropout on the same architecture Table IV Based on the recurrent unit used in our experiments, GRU based hybrid methods with F1-score 95.04 compared with LSTM based hybrid models with F1-score 94.67, GRU units improved the score by 0.37% Table V. Our results show that the hybrid CNN/RNN-based models outperform Bi-dir. Jordan-RNN baseline by 1.13% on the ATIS benchmark Table VI.

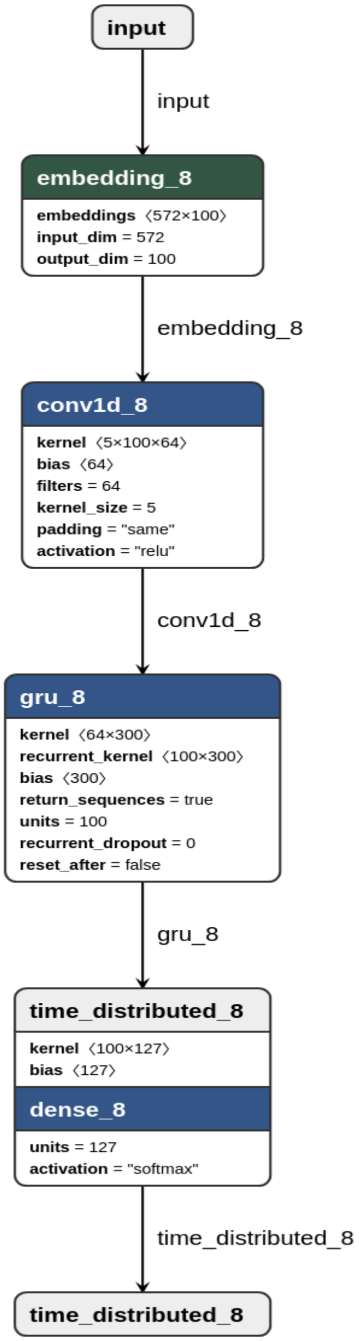

Fig. 6. Best Model Architecture, convolution layer with RNN layer of GRU units without dropout

TABLE IV
COMPARISON BETWEEN HYBRID STRUCTURES BASED ON THE USED RECURRENT UNIT

Structure description	**Precision**	**Recall**	**F1-score**	**Average**
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25; RMSProp	94.47	95.61	95.04	94.89 ±0.15
Hybrid structure Convolution1D and RNN/LSTM with dropout 0.25; RMSProp	94.36	95.17	94.76	94.40 ±0.35

TABLE V
COMPARISON BETWEEN DIFFERENT DEEP LEARNING STRUCTURES BASED ON DROPOUT VALUE

Structure description	**Precision**	**Recall**	**F1-score**	**Average**
Hybrid structure Convolution1D and RNN/GRU without dropout; RMSProp	94.98	95.47	95.11	94.69 ±0.47
Hybrid structure Convolution1D and RNN/GRU with dropout 0.1; RMSProb	94.29	95.31	94.82	94.42 ±0.28
Hybrid structure Convolution1D and RNN/GRU with dropout 0.25; RMSProp	94.47	95.61	95.04	94.89 ±0.15
Hybrid structure Convolution1D and RNN/GRU with dropout 0.5; RMSProp	93.3	94.6	93.95	93.25 ±0.43

TABLE VI
COMPARISON BETWEEN HYBRID STRUCTURES BASED ON THE USED RECURENT UNIT

Models	**Precision**	**Recall**	**F1-score**
Jordan-RNN [18]	92.76	93.87	93.31
Bi-dir. Jordan-RNN [18]	93.82	94.15	93.98
Hybrid structure (Our)	**94.98**	**95.47**	**95.11**

V. CONCLUSION

This paper addresses the problem of slot filling in Spoken Language Understanding. In particular, we focused on slot tagging without paying attention to the other intent classification part. We formulated our learning architecture as a hierarchy of spatial CNN features followed by the RNNs to model dependencies in the temporal domain. Experimental results on the ATIS dataset consistently demonstrated the effectiveness of the proposed approach. It is good to mention that combined models that solve the two tasks at the same time could be implemented and these models had proven to lead to better performance. But still, in the way to implement a full chatbot, we will need to generate human-like text in response to users input. In future work, we intend to explore the incorporation of attentional mechanism in our model, which could provide additional information to the slot label prediction, and learn our architecture using another data-sets to generalize the results.

REFERENCES

[1] P. Su, N. Mrksic, I. Casanueva and I. Vulic, “Deep Learning for Conversational AI NAACL 2018 Tutorial,“ PolyAI, University of Cambridge, 2018

[2] P. Haffner, G. Tur, and J. H. Wright, “Optimizing svms for complex call classification,“ Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP03). 2003 IEEE International Conference on, vol. 1. IEEE, 2003, pp. I632.

[3] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, “Deep belief nets for natural language call-routing,“ Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011, pp. 56805683.

[4] A. McCallum, D. Freitag, and F. C. Pereira, “Maximum entropy markov models for information extraction and segmentation.“ ICML, vol. 17, 2000, pp. 591598.

[5] C. Raymond and G. Riccardi, “Generative and discriminative algorithms for spoken language understanding,“ INTERSPEECH, 2007, pp. 16051608.

[6] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, 2014, pp. 189194.

[7] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. HakkaniTur, X. He, L. Heck, G. Tur and D. Yu et al., "Using recurrent neural networks for slot filling in spoken language understanding," Audio, Speech, and Language Processing, IEEE/ACM Transactions on, vol. 23, no. 3, pp. 530539, 2015.

[8] B. Liu and I. Lane, "Recurrent neural network structured output prediction for spoken language understanding," Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions, 2015.

[9] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, 2014, pp. 554559.

[10] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 7883.

[11] M. Surdeanu and H. Ji, "Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation", 3rd International Workshop on Knowledge Discovery on the WEB, 2017

[12] U. Schade and M. R. Hieb, "Formalizing Battle Management Language:A Grammar for Specifying Orders", 06S-SIW-068 Spring 2006, 2006

[13] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentencelevel information with encoder lstm for natural language understanding," arXiv preprint arXiv:1601.01530, 2016.

[14] S. T. Hsu, C. Moon, P. Jones and N. F. Samatova "A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification," 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017

[15] Y. He and S. Young, "A data-driven spoken language understanding system," in IEEE ASRU 2003.

[16] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in Interspeech 2007

[17] G. Tur, D. Hakkani-Tur, and L. Heck, "What is left to be understood in ATIS¿' in IEEE SLT, 2010

[18] G. Mesnil, X. He, L. Deng and Y.Bengio, "Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding," INTERSPEECH 2013, pp 3771-3775.

[19] B. Qu, X. Li, D. Tao, and X. Lu, "Deep semantic understanding of high resolution remote sensing image," in 2016 International Conference on Computer, Information and Telecommunication Systems (CITS), 2016 .

[20] J. Chung, C. Gulcehre, K. Cho and Y. Bengio "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555 [cs.NE], 2014

[21] C. Sutton, A. McCallum and K. Rohanimanesh, " Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data," JMLR, 2007, pp. 693-723

[22] G. Gao, Y. Hsu, C. Huo, T. Chen, K, Hsu and Y. Che, "Slot-Gated Modeling for Joint Slot Filling and Intent Prediction," Proceedings of NAACL-HLT 2018, pp 753-757