



User Information Augmented Semantic Frame Parsing using Progressive Neural Networks

Yilin Shen, Xiangyu Zeng, Yu Wang, Hongxia Jin

Samsung Research America, Mountain View, CA, USA

{yilin.shen, shane.z, yu.wang1, hongxia.jin}@samsung.com

Abstract

Semantic frame parsing is a crucial component in spoken language understanding (SLU) to build spoken dialog systems. It consists of two main tasks: intent detection and slot filling. State-of-the-art deep learning models have demonstrated good results on these tasks. However, these models require not only a large scale annotated training set but also a long training procedure. In this paper, we aim to alleviate these drawbacks for semantic frame parsing by utilizing the ubiquitous user information. We design a novel progressive deep neural network model to incorporate prior knowledge of user information intermediately to better and quickly train a semantic frame parser. Due to the lack of benchmark dataset with real user information, we synthesize the simplest type of user information (location and time) on ATIS benchmark data. The results show that our approach leverages such simple user information to outperform state-of-the-art approaches by 0.25% for intent detection and 0.31% for slot filling using standard training data. When using smaller training data, the performance improvement on intent detection and slot filling reaches up to 1.35% and 1.20% respectively. We also show that our approach can achieve similar performance as state-of-the-art approaches by using less than 80% annotated training data. Moreover, the training time to achieve the similar performance is also reduced by over 60%.

Index Terms: Spoken Language Understanding, User Information Augmentation, Progressive Neural Networks

1. Introduction

With the emergence of artificially intelligent voice-enabled personal assistants in daily life, spoken language understanding (SLU) system has attracted increasing research attentions. As the key component in a SLU system, semantic frame parsing aims to identify user's intent and extract semantic constituents from a natural language utterance, a.k.a. intent detection and slot filling. Existing approaches includes the independent models for learning intent detection [1, 2] and slot filling [3, 4, 5, 6, 7, 8] separately as well as joint models to learn these two tasks together [9, 10, 11, 2].

Unfortunately, the aforementioned approaches suffer from several main drawbacks. First, they require the existence of a large scale annotated corpus to train a high quality parser. Since a SLU system aims to understand all varieties of user utterances, the corpus is further required to extensively cover all varieties of utterances. However, the collection of such an annotated corpus is very expensive and needs heavy human labor. Secondly, the training of existing parser models oftentimes takes a long time to achieve a good performance. These drawbacks are magnified especially with the recent quick growth of capabilities in personal assistants [12]. To develop a new domain, we need to generate a new utterance dataset and take a long time to

train a new semantic frame parsing model. Thus, it is critically desirable to design a new semantic frame parsing model to alleviate the needs of both large amount of annotated training data and long training time.

In this paper, we investigate how user information can be incorporated into semantic frame parsing to overcome the above drawbacks. We design a novel progressive attention-based recurrent neural network (Prog-BiRNN) model that first annotates the information types and then distills the related prior knowledge w.r.t. each type of information to continue learning intent detection and slot filling. Our approach is motivated by the recent success of attention-based RNN model [2] for joint learning of intent detection and slot filling and progressive neural networks [13] in many multi-tasking learning applications. Our model includes a main RNN structure stacked with a set of different layers and they are trained one by one in a progressive manner.

Organization: Section 2 describes the background and related work. We discuss our new problem definition in Section 3. Section 4 includes our proposed model and its training procedure details. We show the experimental results in Section 5. Section 6 concludes the whole paper.

2. Background & Related Work

2.1. Semantic Frame Parsing

Intent detection and slot filling are two main tasks to build a semantic frame parser for spoken language understanding (SLU). That is, the goal of semantic frame parsing is to understand all varieties of user utterances by correctly identifying user's intents and slot tags. Given an input utterance as a sequence \mathbf{x} of length T , intent detection identifies the intent class I for \mathbf{x} and slot filling maps \mathbf{x} to the corresponding label sequence \mathbf{y} of the same length T (Table 1).

Intent detection is treated as an utterance classification problem, which can be modeled using conventional classifiers such as support vector machine (SVM) [1] and RNN based models [2]. As a sequence labeling problem, slot filling can be solved using traditional machine learning approaches including maximum entropy Markov model [3] and conditional random fields (CRF) [14], as well as recurrent neural network (RNN) based approaches which takes and tags each word in an utterance one by one [4, 5, 6, 7, 8]. Recent research focuses on the joint model to learn two tasks together [9, 10, 11, 2].

2.2. Joint Attention-based RNN Model

We recall the state-of-the-art approach in [2], referred to as *Att-BiRNN* model, which will be used as the base of our approach. Att-BiRNN is a joint RNN model to learn the two tasks together. It first uses a bidirectional RNN with a basic LSTM cell to read the input utterance as a sequence \mathbf{x} . At each time stamp t , a

context vector \mathbf{c}_t is learned to concatenate with the RNN hidden state \mathbf{h}_t , i.e., $\mathbf{c}_t \oplus \mathbf{h}_t$, to learn a slot attention for predicting the slot tag y_t . All hidden states of slot filling attention layer are used to predict the intent label in the end. The objective function of Att-BiRNN model is as follows:

$$P(\mathbf{y}|\mathbf{x}) = \max_{\theta_r, \theta_s, \theta_I} \prod_{t=1}^T P(y_t|y_1, \dots, y_{t-1}, \mathbf{x}; \theta_r, \theta_s, \theta_I) \quad (1)$$

where $\theta_r, \theta_s, \theta_I$ are the trainable parameters of different components (utterance BiRNN, slot filling attention layer and intent classifier) in Att-BiRNN model.

3. Problem Definition

We propose the *User Info Augmented Semantic Frame Parsing* problem for the same two tasks, intent detection and slot filling, by considering the following additional inputs.

User Info Dictionary: This defines the categorical relations between user info type and slots. In other words, each key in the dictionary is a type of user info and its corresponding value is the slots belonging to this type. The generation of this dictionary is not the focus of our paper since it can be simply generated by a software developer when he generates slots during the development of a new domain in practice.

Each type of user info is associated with an external or pre-trained model to extract their semantically meaningful prior knowledge. For example, the semantics of a location is represented by its longitude and latitude such that the distance between two locations reflect their actual geographical distance.

User Info for Each Utterance: Each input sequence \mathbf{x} is associated with its corresponding user info U . U is represented as a set of tuples, $\langle \text{Info Type}, \text{Info Content} \rangle$. As an example utterance in Table 1, the first gray row shows our generated user info with type “User Location” and content “Brooklyn, NY”. Learning user info has been well studied, such as user contextual information (e.g., time, location, activity, etc.) via smartphone [15], Internet of Things [16] and user interests (e.g., favorite food, etc.) using recommendation models [17].

Remarks: One may argue that this is a simple extension of semantic frame parsing problem in which the user info can be simply encoded into an existing model as a new input or a new state. However, these naive approaches ignore the different semantic meanings between user info and language context in an utterance, as well as between different types of user info. Thus, as we later show in experiment (Section 5), these baseline approaches do not show any advantage over existing approaches without user info.

4. Proposed Approach

In this section, we describe the main idea and details of our proposed Prog-BiRNN model as well as its training procedure.

4.1. Progressive Attention-based RNN Model

As the name indicates, our main idea is to train the semantic frame parsing model progressively with an intermediate task before achieving the final goal of intent detection and slot filling. This is motivated by the recent success of progressive neural networks [13]. Specifically, for each utterance \mathbf{x} , we first define the user info sequence \mathbf{z} using the user info dictionary. In Table 1, the last row shows the user info sequence corresponding to this example. Our approach first trains a user info tagging to derive \mathbf{z} . Then, the prior knowledge with

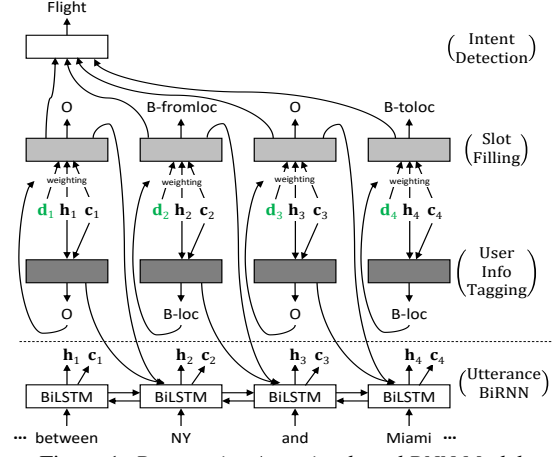


Figure 1: Progressive Attention based RNN Model

semantic meaning for each type of user info is distilled into the model to continue training for intent detection and slot filling.

Table 1: ATIS corpus sample with intent and slot annotations with additional user info and its corresponding user info sequence (in gray)

utterance (\mathbf{x})	round	trip	flights between	ny	and	miami
slots (\mathbf{y})	B-round,trip	I-round,trip	O	O	B-fromloc	O B-toloc
intent (I)	atis.flight					
user info (U)	{“User Location”: “Brooklyn, NY”}					
user info seq (\mathbf{z})	O	O	O	O	B-loc	O B-loc

As shown in Figure 1, our proposed Prog-BiRNN model is designed based on the state-of-the-art Att-BiRNN model [2], which consists of the following four main components.

Utterance BiRNN Layer: We use the same bidirectional RNN (BiRNN) to encode an utterance with LSTM cells (BiLSTM) as in [2]. The hidden state \mathbf{h}_t at each time step t is the concatenation of forward state \mathbf{fh}_t and backward state \mathbf{bh}_t , i.e., $\mathbf{h}_t = \mathbf{fh}_t \oplus \mathbf{bh}_t$.

User Info Tagging Layer: This component labels the user info type for each word in the input utterance. Since the labeling is based on the language context of input utterance, we follow the previous work [2] to use a language context vector \mathbf{c}_t at each time stamp t via the weighted sum of all hidden states $\{\mathbf{h}_k\}_{\forall 1 \leq k \leq T}$ i.e., $\mathbf{c}_t = \sum_{k=1}^T \alpha_{t,k} \mathbf{h}_k$. Here, $\alpha_t = \text{softmax}(\mathbf{e}_t)$, i.e., $\alpha_{t,j} = \frac{\exp(e_{t,j})}{\sum_{k=1}^T \exp(e_{t,k})}$. $e_{t,k} = g(\mathbf{s}_{t-1}^u, \mathbf{h}_k)$ is also learned from a feed forward neural network g with the previous hidden state \mathbf{s}_{t-1}^u defined as the concatenation of \mathbf{h}_{t-1} and \mathbf{c}_{t-1} , i.e., $\mathbf{s}_{t-1}^u = \mathbf{h}_{t-1} \oplus \mathbf{c}_{t-1}$. At each time step t , the user info tagging layer outputs $P_u(t)$ as follows:

$$P_t^u = \text{softmax}(\mathbf{W}^u \mathbf{s}_t^u); \quad \hat{z}_t = \arg \max_{\theta_u} P_t^u \quad (2)$$

Slot Filling Layer: This is the key layer for distilling user info into the model to help reduce the need of annotated training data. It shares the same hidden state \mathbf{h}_t and language context \mathbf{c}_t with the user info tagging layer. For each word in the utterance, we use external knowledge to derive the prior distance vectors $\mathbf{d}_t = \{\mathbf{d}_t(1), \dots, \mathbf{d}_t(|U|)\}$ for each time stamp t (green in Figure 1) where $|U|$ is the number of user info types in IOB format. And each element \mathbf{d}_t^j is defined as follows:

$$\mathbf{d}_t(j) = \text{sigmoid}(\beta(j) \odot \delta_t(j)) \quad (3)$$

where \odot stands for element-wise multiplication. $\beta(j)$ is a $|U|$

dimensional trainable vector; and $\delta_t(j)$ is the distance between the t^{th} word and user info w.r.t. the prior knowledge of type j .

Next, we define the calculation of distance $\delta_t(j)$ for each info type j at time stamp t , through the example in Figure 1. Let $\delta_t(\text{loc})$ be the distance w.r.t. the location type of user info. It is a one-dimensional scalar in this case. Taking the second word “NY” as an example, we have its following location distance since it is tagged as “Location” type of user info:

$$\delta_2(\text{loc}) = \text{dist}(\text{“NY”}, \text{“Brooklyn, NY”}) \approx 4.8 \text{ (miles)}$$

by using external location based services, i.e., Google Maps Distance Matrix API [18]. If the word and user info are of different types, we set the distance $\delta_t(j)$ as -1 such that its corresponding $d_t(j)$ will be close to 0 via the sigmoid function.

To feed the prior distance vectors \mathbf{d}_t into the slot filling layer, we weight each element $\mathbf{d}_t(j)$ and the language context \mathbf{c}_t over the softmax probability distribution P_t^u from the user info tagging layer. Intuitively, this determines how important a type of user info or the language context in utterance is to predict the slot tag of each word in the utterance. Thus, we have the input Φ_t of LSTM cell at each time step t in slot filling layer as follows:

$$\Phi_t = P_t^u(1)\mathbf{d}_t(1) \oplus \dots \oplus P_t^u(|U|)\mathbf{d}_t(|U|) \oplus P_t^O \mathbf{c}_t \quad (4)$$

where $P_t^u(j)$ and P_t^O stand for the probability that the t^{th} word is predicted as j type of user info and as “O” meaning none of the types. Note that we will discuss how to deal with IOB format in Section 4.2.2. At last, the state \mathbf{s}_t^s at time step t is computed as $\mathbf{h}_t \oplus \Phi_t$ and the slot tag is predicted as follows:

$$P_t^s = \mathbf{W}^s \mathbf{s}_t^s; \quad \hat{y}_t = \arg \max_{\theta_s} P_t^s \quad (5)$$

Intent Detection Layer: We add an additional intent detection layer as in [2] to generate the probability distribution P_I of intent class labels by using the concatenation of hidden states from slot filling layer, i.e., $\mathbf{s}^I = \mathbf{s}_1^I \oplus \dots \oplus \mathbf{s}_T^I$.

$$P^I = \text{softmax}(\mathbf{W}^I \mathbf{s}^I); \quad \hat{I} = \arg \max_{\theta_I} P^I$$

Remarks: The sharing of hidden state \mathbf{h}_t and language context \mathbf{c}_t between user info tagging and slot filling layers is crucial to reduced the required annotated training data. For the user info tagging layer, $\mathbf{h}_t, \mathbf{c}_t$ are mainly used to tag the words which belong to one type of user info. The semantic slots of these words can be easily tagged in slot filling layer by utilizing the distilled prior knowledge instead of using $\mathbf{h}_t, \mathbf{c}_t$ again. The slot filling then depends on $\mathbf{h}_t, \mathbf{c}_t$ to tag the rest of words not belonging to any type of user info.

4.2. Progressive Training with IOB Format Support

4.2.1. Training Algorithm

The training procedure is progressively conducted step by step. The first step is to train user info tagging component with loss function \mathcal{L}_u as follows:

$$\mathcal{L}_u(\theta_r, \theta_u) \triangleq -\frac{1}{n} \sum_{i=1}^{|U|} \sum_{t=1}^n z_t(i) \log P_t^u(i) \quad (6)$$

where $|U|$ is the number of user info types in IOB format.

Then, we train the slot filling layer with loss function \mathcal{L}_s and intent classifier with loss function \mathcal{L}_I simultaneously. In the meanwhile, we also allow the fine tuning of parameters θ_r and θ_u in utterance BiRNN and user info tagging layers.

$$\mathcal{L}_s(\theta_r, \theta_I, \theta_s, \theta_u) \triangleq -\frac{1}{n} \sum_{i=1}^{|S|} \sum_{t=1}^n y_t(i) \log P_t^s(i) \quad (7)$$

$$\mathcal{L}_I(\theta_r, \theta_I, \theta_s, \theta_u) \triangleq -\sum_{i=1}^{|I|} I(i) \log P^I(i) \quad (8)$$

where $|S|$ is the number of slots in IOB format and $|I|$ is the number of intents. $P(i)$ stands for the probability $P(X = x_i)$. Moreover, $\theta_r, \theta_u, \theta_s, \theta_I$ are the parameters in utterance BiRNN, user info tagging, slot filling and intent detection components in our proposed Prog-BiRNN model.

4.2.2. Details of IOB Format Support

Thanks to the progressive training procedure, the IOB format will be naturally supported in our model. As shown in Figure 2, in the case of “New York” with “B-loc I-loc” user info tags, we take them together to extract the prior geographical distance $\text{dist}(\text{“New York”}, \text{“Brooklyn, NY”})$. Moreover, since B-loc and I-loc are considered as different tags in the output P_t^u of user info tagging component, they can be directly used to infer B-fromloc and I-fromloc in slot filling component accordingly.

In the case that the type of user info for the t^{th} word is incorrectly tagged, the hidden state \mathbf{h}_t and language context \mathbf{c}_t will be used to infer the slot tags since the user info tagging output P_t^u will weight more on $\mathbf{h}_t, \mathbf{c}_t$ in this case. In addition, the second phase of training procedure for joint training of all components also leans to use more language context to correct the incorrectly tagged type of user info.

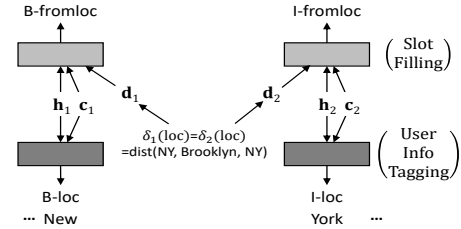


Figure 2: Support of IOB Format (omitted other model details)

Remarks: The capability of prior knowledge distillation in our approach leverages user information to largely improve the performance and reduce the requirement of annotated training data. Moreover, the overall training time is also largely shortened since our approach divides SLU into simpler subproblems in which each subproblem is much easier to train.

5. Experimental Evaluation

5.1. Dataset

We evaluate our approach on the ATIS (Airline Travel Information Systems) dataset [19], a widely used dataset in SLU research. The training set contains 4,978 utterances from the ATIS-2 and ATIS-3 corpora, and the test set contains 893 utterances from the ATIS-3 data sets. There are 127 distinct slot labels and 22 different intent classes.

Due to the lack of benchmark datasets with user info, we design the following two mechanisms to synthesize two types of user info, *user contextual location* and *user preferred time periods* in ATIS dataset. We first construct the user info dictionary by including all slots with “loc” keyword in contextual location and including all slots with “time” keyword in user preferred time period.

The prior distance δ of contextual location are computed using Google Maps Distance Matrix API [18]. For time period, we calculate δ by using the difference between the tagged time stamp in an utterance and the middle time stamp of the user preferred time period.

Contextual Location: W.l.o.g., we synthesize user contextual

Table 2: Examples of synthesized user info in ATIS dataset

Utterance	User Info	
	Type	Content
i need a flight from dallas to san francisco {“fromloc.city_name”: “dallas”}	contextual location	Fort Worth,TX
all flights to baltimore after 6 pm {“depart.time.time”: “6 pm”}	preferred depart period	evening
i want to fly from boston at 838 am and arrive in denver at 1110 in the morning {“fromloc.city_name”: “boston”} {“arrive.time.time”: “1110”}	contextual location	Cambridge,MA
{“arrive.time.period_of_day”: “morning”}	preferred arrive period	morning

locations based on the intuitive assumption that user’s location is usually close to flight depart city. We first extract all values (real locations) of slots which contains “fromloc” in their names. Then, for each real location, we use Google Places API [20] to find the nearby cities within 50 km. For each utterance having slots containing ‘fromloc’, we add the nearby city of this slot value as its location. When there are more than one nearby cities, we randomly select one from them.

Preferred Time Periods: We follow Oxford dictionary to consider four periods of a day: morning (6am-12pm), afternoon (12pm-6pm), evening (6pm-12am), night (12am-6am). In each utterance having the slots with “time” keyword, we generate one depart and one arrive time preference by selecting from these four periods as follows: If there is a slot containing ‘depart.time’, we set the preferred time period based on the value of this slot. For example, if the slot value is “8pm”, we set the preferred time period to be “evening” since “8pm” belongs to the period 6pm-12am. For the slot ‘depart.time.period_of_day’, we simply match the key words to synthesize the user preferred depart time period. We synthesize the arrive period preference in the same way.

5.2. Baseline Competitors & Implementation Details

In addition to the state-of-the-art baseline Att-BiRNN in [2], we also design another baseline competitor using user info as discussed at the end of Section 3. For the sake of fairness, we consider concatenating the user info directly to the input of slot filling layer in the Att-BiRNN. All user info is concatenated together without distinguishing different types. We call these two baselines *Att-BiRNN with/without User Info* respectively.

Also, we follow the exact same hyperparameters in the original paper of the base Att-BiRNN model [2] since our model does not have additional hyperparameters.

5.3. Results with Different Sizes of Training Set

We evaluate our Prog-BiRNN model on subsets of full size ATIS training set and randomly sampled 3 different sizes (2,000, 3,000 and 4,000) utterances out of the total 4,978 utterances. Figure 3 reports the average performance results on 10 differently sampled training set of each size.

Since location related are the majority of all slots in ATIS dataset, we first consider only using contextual location as user info. As shown in Figure 3a, the F1 score of slot filling outperforms both baseline approaches with around 0.2% absolute gain of each size. The accuracy improvement of intent detection is around 0.1% and up to 0.2% for full size training set. This slightly smaller improvement margin is due to the small number of intent classes. When using both contextual location and preferred time period as user info, we observe more significant improvement with 0.25% gain for intent detection and 0.31% gain for slot filling. Note that

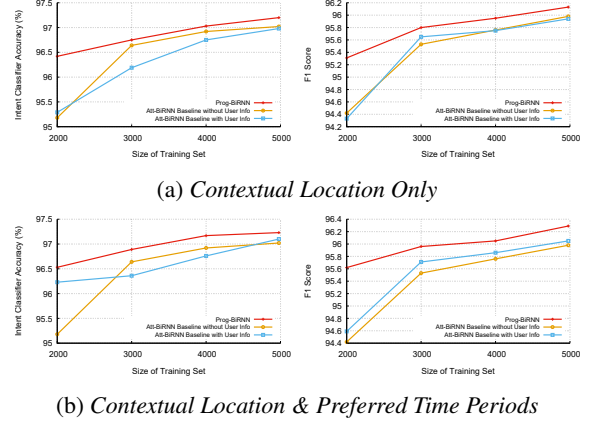


Figure 3: Performance results with different sizes of training set

our reported intent detection accuracy is different from that in baseline paper [2] since we use all 22 intents in ATIS dataset. In particular, when using smaller training data, i.e., 2000 training data, the performance improvement on intent detection and slot filling reaches 1.35% and 1.20% respectively. More significantly, our Prog-BiRNN model can use less than 4000 (80%) annotated utterances with simple user location and preferred time period as training data to achieve the performance of baseline approaches for both intent detection and slot filling.

5.4. Training Time Results

We also report the training time between our Prog-BiRNN and baseline approaches. Since our approach mainly focuses on improving slot filling, Figure 4 reports the averaged slot filling F1 score after each epoch of training. Thanks to the small number of user info types, the first user info tagging training phase only takes 3 epochs to achieve over 92% accuracy, which is sufficient for the second training phase. As one can see, the number of epochs (3 epochs included) takes to achieve a competitive performance of slot filling is around over 60% smaller than both two baseline approaches.

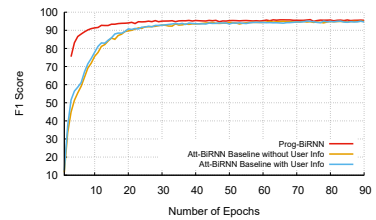


Figure 4: Training time results on full size training set using both contextual location & preferred time periods as user info

6. Conclusion

We present a novel progressive neural network model to train a semantic frame parsing model by incorporating user information. By using simple user information, we show that our approach not only significantly improves the performance but largely reduces the needs of annotated training set as well. In addition, our approach also shows its ability to shorten the training time for achieving the competitive performance. Thus, we enable the quick development of a semantic frame parsing model with less annotated training set in new domains.

7. References

- [1] P. Haffner, G. Tur, and J. H. Wright, "Optimizing svms for complex call classification," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 1–1.
- [2] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *arXiv preprint arXiv:1609.01454*, 2016.
- [3] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 591–598.
- [4] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 189–194.
- [5] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "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.
- [6] B. Peng and K. Yao, "Recurrent neural networks with external memory for language understanding," *arXiv preprint arXiv:1506.00195*, 2015.
- [7] B. Liu and I. Lane, "Recurrent neural network structured output prediction for spoken language understanding," in *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*, 2015.
- [8] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentencelevel information with encoder lstm for natural language understanding," *arXiv preprint*, 2016.
- [9] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 554–559.
- [10] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec 2013, pp. 78–83.
- [11] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *INTERSPEECH*, 2016, pp. 715–719.
- [12] <http://www.businessinsider.com/amazon-alexa-how-many-skills-chart-2017-7>.
- [13] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.
- [14] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *INTERSPEECH*, 2007.
- [15] . Yürür, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung, "Context-awareness for mobile sensing: A survey and future directions," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 68–93, 2016.
- [16] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [17] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, vol. 2009, pp. 4:2–4:2, Jan. 2009. [Online]. Available: <http://dx.doi.org/10.1155/2009/421425>
- [18] <https://developers.google.com/maps/documentation/distance-matrix>.
- [19] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '90. Stroudsburg, PA, USA: Association for Computational Linguistics, 1990, pp. 96–101.
- [20] <https://developers.google.com/places/web-service>.