Using Bidirectional Recurrent Neural Networks for Slot Filling in Spoken Language Understanding

Sotirios S. Dimitras, *2013030123*

***Abstract*—Semantic slot filling is one of the most challenging problems in spoken language understanding (SLU). In this paper, we propose to use bidirectional recurrent neural networks (Bi-dir RNNs) for this task, and present an architecture designed to efficiently model past and future temporal dependencies. Specifically, we implemented and compared Bidirectional RNNs using the Basic RNN Cell and the Long Short-Term Memory (LSTM) Cell. We implemented these networks with the publicly available Tensorflow neural network toolkit and completed experiments on the Airline Travel Information System (ATIS) datasheet. In addition, we compared our approaches' results, with those of the Elman Forward architecture, which was given to us. It is worth nothing that Elman Forward architecture was implemented with Theano neural network toolkit. Our results show that the Biderectional RNN-based models outperformed the Elman Forward architecture 's results by 3% on the ATIS benchmark..**

***Index Terms*—**spoken language understanding, natural language processing, word embedding, recurrent neural network, slot filling.

I. Introduction

Spoken language understanding (SLU) system is a critical component in spoken dialogue systems. SLU system typically involves identifying speaker's intent and extracting semantic constituents from the natural language query, two tasks that are often referred to as intent detection and slot filling.

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) and deep neural network methods can be applied. Slot filling is the task to assign a semantic concept to each word in a sentence. For example, in the sentence 'I want to fly from Athens to Chania', an SLU system should tag 'Athens' as the departure city of a trip and 'Chania' as the arrival city. All the other words, which do not correspond to real slots, are then tagged with an artificial class O. Popular approaches to solving the slot filling task in SLU include maximum entropy Markov models (MEMMs), conditional random fields (CRFs), and recurrent neural networks (RNNs). This project will focus on the use of RNNs.

RNNs have demonstrated to be successful in many natural language processing tasks, such as language modeling, language understanding and machine translation. A simple RNN consists of an input layer, a recurrent hidden layer, and an output layer. The input layer reads each word and the output layer produces probabilities for the target labels. These can be words for language modeling and machine translation or semantic labels for the slot filling task. The network can be trained with backpropagation through time and therefore can save information from the input words of several time steps to make the prediction for the current word.

In this project, we apply a bidirectional recurrent neural netwrok to the slot filling task, using two approaches. The first approach uses a Vanilla RNN Cell, and the second uses a LSTM RNN Cell. For each input word, the network take the information not only from the previous words but also from the future words to predict the target slot. Furthermore instead of using cross entropy loss to train the models, we use a cross entropy loss function. Finally, we test both of the models on the ATIS benchmark data set and show that our models outperform the Elman Forward architecture results, using no additional knowledge or data sources.

II. Slot Filling In Natural Language Understanding

Slot filling can be treated as a sequence labeling problem, where we have training examples of $\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}) : n = 1, .., N\}$ and we want to learn a function $f : X \rightarrow Y$ that maps an input sequence x to the corresponding label sequence y. In slot filling, the input sequence and label sequence are of the same length, and thus there is explicit alignment.

An example sentence is provided here, with domain, intent, and slot/concept annotations illustrated, along with typical domain-independent named entities. This example follows the popular in/out/begin (IOB) representation, where Boston and New York are the departure and arrival cities specified as the slot values in the user's utterance, respectively.

Sentence	*show*	*flights*	*from*	*Boston*	*To*	*New*	*York*	*today*
Slots/Concepts	O	O	O	B-dept	O	B-arr	I-arr	B-date
Named Entity	O	O	O	B-city	O	B-city	I-city	O
Intent	*Find_Flight*							
Domain	*Airline Travel*							

ATIS utterance example IOB representation

RNNs have been widely used in many sequence modeling problems. At each time step of slot filling, RNN reads a word as input and predicts its corresponding slot label considering all available information from the input and the emitted output sequences. The model is trained to find the best parameter set θ that maximizes the likelihood:

$$\arg\max_{\theta} \prod_{t=1}^{T} P(y_t | y^{(t-1)}, x; \theta)$$

where x represents the input word sequence, y_1^{t-1} represents the output label sequence prior to time step t. During inference, we want to find the best label sequence y given an input sequence x such that:

$$\hat{y} = \arg\max_{y} P(y|x)$$

III. Recurrent Neural Networks For Slot Filling

A. RNN Inputs

As input for the RNN, we use word embeddings which are randomly initialized and jointly trained with the network. Initial experiments showed that concatenating embeddings of tri grams instead of using single words leads to superior results. Hence at time-step t, we do not only give the embedding of word w_t to the model but the concatenated embeddings of the trigram $w_{t-1}w_tw_{t+1}$. In the following figure, however, we depict a standard RNN input (single words) to avoid distractions from the focus of the pictures which is the directionality and not the input of the models

B. Bi-Directional RNNs

Especially for slot filling, the processing of the slot arguments might be easier with knowledge of the succeeding words. Therefore in bi-directional RNNs, not only the previous history of word wt is regarded but also the future history. Figure 1 shows this in which U_b and U_f are shared and then denoted with U in the equation 1. Thus, the network can be split into three parts: a forward pass which processes the original sentence word by word; a backward pass which processes the reversed sentence word by word; and a combination of both. All three parts are trained jointly. The combination can be done by adding the forward and the backward hidden layer. This leads to the following hidden layer output at time step t:

$$h_t = f\left(U w_t + V_b h_{b_{t+1}} + V_f h_{f_{t-1}}\right) \quad (1)$$

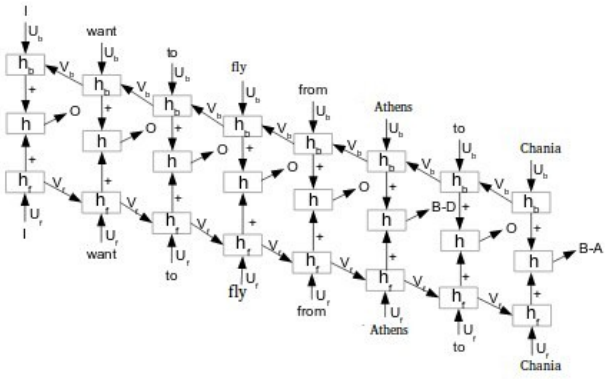

Fig. 1. Bi-directional RNN for slot filling task

C. Word Embeddings

As an alternative to N-gram models, researchers came up with several different techniques based on learning Euclidean space tructures for words. A real-valued embedding vector is associated with each word, and these embeddings are usually trained in an unsupervised way on a large corpus of natural language,e.g. Wikipedia. The architecture of these models can vary from shallow neural nets (NN) or convolutional nets (SENNA) to recurrent neural nets (RNN). The learned word embedding shows good generalization properties across many common natural language processing (NLP) tasks. The neural network architectures evaluated in this paper are based on such word embeddings.

D. Two Types Of RNN Cells

In our project, we had to implement two models using two different cells. The first model should be built using a basic RNN cell (Vanilla Cell), and the second model should be built using the LSTM cell (Long Short-Term Memory). The main difference from a simple RNN is that the LSTM uses a memory cell with linear activation function to store information.

IV. Experiments

A. Data Set

Dataset we are going to use is Airline Travel Information System (ATIS). This dataset was collected by DARPA in the early 90s. ATIS consists of spoken queries on flight related information. The ATIS official split contains 4,978/893 sentences for a total of 56,590/9,198 words (average sentence length is 15) in the train/test set. The number of classes (different slots) is 128 including the O label (NULL). Unseen words in the test set are encoded by <UNK> token and each digit is replaced with string DIGIT, i.e 20 is converted to DIGITDIGIT

B. Training Procedure

We used the Tensorflow library to implement each model. To train the model, each sentence was passed as a batch to the model. We used all of the 5 ATIS folds which were given to us. We trained and evaluated our models for each fold separately.

Word embeddings of size 64 are randomly initialized and fine-tuned during mini-batch training with batch size of 16. Dropout rate 0.5 is applied to the non-recurrent connections during model training for regularization.

C. Results

In this experiment, we compare the uni-directional (Elman Forward Architecture) and bi-directional RNN trained with the cross-entropy loss function. Since the information of the future words can be used to predict the semantic label of the target word, we also report the F1-score by using only the backward pass. The results listed at the Table 1 reveal that the bi-directional RNN outperforms the uni-directional one.

	F1-score				
Model	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4
Elman -Forward	94.45	94.06	93.74	94.68	94.11
Bi-Dir RNN (Vanilla)	97.57	97.68	97.62	97.64	97.60
Bi-Dir RNN (LSTM)	97.47	97.55	97.48	97.27	97.41

Table 1: uni- vs. bi-directional RNN with cross-entropy loss function

V. Conclusions

In this paper, we presented two bi-directional recurrent neural networks for the slot filling task of spoken language

understanding. These networks take the information not only from past but also from future contexts to predict the semantic label of the target word. The results revealed that integrating future information is important for this task. Furthermore, using a ranking loss function to train the model improved the performance over the cross entropy loss function. On the ATIS spoken language understanding task, we achieved new state-of-the-art results with 97.68% macro F1-score for Basic RNN and 97.55% for LSTM RNN without using any additional features or data sources.

REFERENCES

[1] Tom Hope, Yehezkel S. Resheff and Itay Lieder "Learning TensorFlow, A Guide to Building Deep Learning Systems", 1st ed., 2017

[2] Sepp Hochreiter and Jürgen Schmidhuber ,"Long Short-Term Memory", Volume 9, Issue 8, 1997

[3] JNgoc Thang Vu,Pankaj Gupta, Heike Adel and Hinrich Schutze, "BI-DIRECTIONAL RECURRENT NEURAL NETWORK WITH RANKING LOSS FOR SPOKEN LANGUAGE UNDERSTANDING"

[4] Grégoire Mesnil, Xiaodong He, Li Deng and Yoshua Bengio, "Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding"

[5] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig, "Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding"

[6] Bing Liu and Ian Lane, "Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling"

[7] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, "SPOKEN LANGUAGE UNDERSTANDING USING LONG SHORT-TERM MEMORY NEURAL NETWORKS"

[8] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation"

[9] Rui Meng, "A Brief Review of Neural Network on Spoken Language Understanding", Ph.D. student, University of Pittsburgh,2016