Improving Slot Filling in Spoken Language Understanding with Joint Pointer and Attention

Lin Zhao and **Zhe Feng**
Bosch Research and Technology Center
Sunnyvale, CA 94085, USA
{lin.zhao, zhe.feng2}@us.bosch.com

Abstract

We present a generative neural network model for slot filling based on a sequence-to-sequence (Seq2Seq) model together with a pointer network, in the situation where only sentence-level slot annotations are available in the spoken dialogue data. This model predicts slot values by jointly learning to copy a word which may be out-of-vocabulary (OOV) from an input utterance through a pointer network, or generate a word within the vocabulary through an attentional Seq2Seq model. Experimental results show the effectiveness of our slot filling model, especially at addressing the OOV problem. Additionally, we integrate the proposed model into a spoken language understanding system and achieve the state-of-the-art performance on the benchmark data.

1 Introduction

Slot filling is a key component in spoken language understanding (SLU), which is usually treated as a sequence labeling problem and solved using methods such as conditional random fields (CRFs) (Raymond and Riccardi, 2007) or recurrent neural networks (RNNs) (Yao et al., 2013; Mesnil et al., 2015).

Although these models have achieved good results, they are learned on the datasets with word-level annotations, e.g., with the BIO tagging schema as in ATIS (Hemphill et al., 1990). Manual annotations at word level require big effort and some corpora has only sentence-level annotations available, e.g., the utterance *"... moderately priced restaurant"* has a slot-value pair annotation of *"pricerange=moderate"*. As such datasets lack explicit alignment between the annotations and the input words, some systems rely on handcrafted rules to find the alignments in order to automatically create word-level labels to learn the sequence model (Zhou and He, 2011; Henderson, 2015), but finding such alignments is non-trivial. For example, it was shown in (Henderson, 2015) that when applying the manually created word aliases to the speech recognition hypotheses, only around 73% of alignments can be found due to the noise, and a CRF model trained on such noisy data performs particularly worse than some other methods. In addition it is time-consuming to adapt the manual rules or aliases to new domains.

Some other work avoids this issue by regarding slot filling as a classification task (Henderson et al., 2012; Williams, 2014; Barahona et al., 2016), where an utterance is classified into one or more slot-value pairs. This, however, brings other challenges. One is that some types of slots may have a large or even unlimited number of possible values, so the classifiers may suffer from the data sparsity problem when the training data is limited. Another is the OOV problem caused by unknown slot values (e.g., restaurant name, street name), which is impossible to predefine and is very common in real-world spoken dialogue applications.

To address these challenges, we present a neural generative model for slot filling on unaligned dialog data, specifically for slot value prediction as it has more challenges caused by OOV. The model uses Seq2Seq learning to predict a sequence of slot values from an utterance. Inspired by the ability of pointer network (Ptr-Net) (Vinyals et al., 2015) at addressing OOV problems, we incorporate Ptr-Net into a standard Seq2Seq attentional model to handle OOV slots. It can predict slot values by either generating one from a fixed vocabulary or selecting a word from the utterance. The final model is a weighted combination of the two operations.

To summarize, our main contributions are:

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 426–431
Melbourne, Australia, July 15 - 20, 2018.

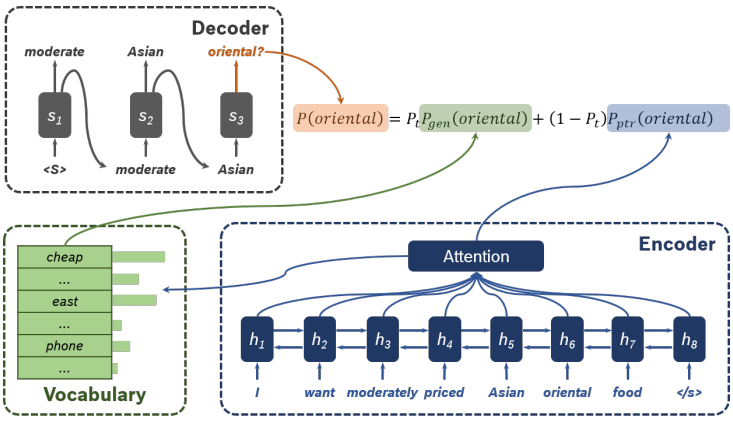

Figure 1: Our model for slot value prediction based on Seq2Seq learning with attention and Ptr-Net.

- We use a neural generative model for slot filling on the data without word-level annotations which has received less attention.
- We adopt the pointer network to handle the OOV problem in slot value prediction, which achieves good performance without any manually-designed rules or features.

2 Background of Pointer Network

Ptr-Net is a variation of the standard Seq2Seq model with attention. At each decoding step, it selects a position from the input sequence based on the attention distribution instead of generating a token from the target vocabulary. Given the input $X = \{x_1, ..., x_T\}$, the output y_t at time step t is predicted by:

$$P_{ptr}(y_t = w | y_1^{t-1}, X) = \sum_{i:x_i=w} a_i^t \qquad (1)$$

where a_i^t is the attention weight of position i at step t. The advantage of Ptr-Net is that it can make better predictions on unknown or rare words. It has been successfully applied to tasks such as abstractive summarization (See et al., 2017), question answering (He et al., 2017), reading comprehension (Wang and Jiang, 2016), and chunking (Zhai et al., 2017).

3 Model for Slot Value Prediction

Our model for slot value prediction is a hybrid of a Seq2Seq attentional model and a Ptr-Net, similar as the one in See et al. (2017). The input is a sequence of words in an utterance, and the output is a sequence of slot values whose tokens may or may not appear in the input.

The hybrid model, illustrated in Figure 1, allows us to both generate a slot value from a fixed vocabulary and pick a value from the input via pointing. The two components (*Seq2Seq* and *Ptr-Net*) share the same encoder-decoder architecture and attention scores. We adopt a single-layer bidirectional GRU (Cho et al., 2014) for the encoder, and a single-layer unidirectional GRU for the decoder. The attention is calculated as in Bahdanau et al. (2014).

The slot vocabulary is set to contain only the values of enumerable slots, but not those of non-enumerable slots (e.g., values of *"restaurant name"*) as we assume these are not known in advance in the real scenarios.

We use the term "extended vocabulary" to denote the union of the slot vocabulary and all words from the input utterances. The probability distribution over the extended vocabulary is calculated as:

$$P(w) = p_t P_{gen} + (1 - p_t) P_{ptr} \qquad (2)$$

That is, the model makes the final predictions using a weighted combination of the predictions from two individual components. At the decoding step t, the Seq2Seq component produces the probability distribution P_{gen} for the next slot value within the vocabulary, while Ptr-Net produces the probability distribution P_{ptr} over the input positions. $p_t \in [0, 1]$ is a parameter to balance the two components. It is learned at each time step based on the decoder input d_t, decoder state s_t and the context vector c_t as follows:

$$p_t = \sigma(w_c c_t + w_s s_t + w_d d_t + b) \qquad (3)$$

where σ is a sigmoid function. w_c, w_s and w_d are all trainable weights.

Model	P	R	F
CNN	**93.3**	76.3	84.0
Seq2Seq w/ attention	86.6	**81.9**	84.2
Our model	88.8	81.3	**84.9**

Table 1: Results of slot value prediction.

4 Experiments

In this section, we present our experimental results on DSTC2 (Dialog State Tracking Challenge) (Henderson et al., 2014), including the results of slot value prediction solely and a complete SLU system. Our models are implemented using Keras[1] with TensorFlow as backend. In all the experiments, the dimension of hidden states is 128, dimension of word embeddings is 100, dropout rate is 0.5, and batch size is 32. Word embeddings are not pre-trained but learned from scratch during training. Teacher forcing is used during training, with Adam optimizer (Kingma and Ba, 2014). All training consists of 10 epochs with early stopping on the development set.

4.1 Data

DSTC2 consists of multi-turn dialogues between users and a dialog system, in the restaurant search domain. Each utterance is annotated with semantics including dialog-acts and slot-value pairs. For an utterance, both its transcription and 10-best hypotheses are provided. We use the top hypothesis as input throughout our experiments. The corpus has been separated into training, development and testing, containing 11,677, 3,934 and 9,890 utterances respectively.

4.2 A Complete SLU System

For better evaluation and comparison, we integrated our model of slot value prediction into a complete SLU system, which uses a CNN classifier to obtain dialog-acts and slot types respectively after slot value prediction. For dialog act prediction, the input to the CNN model is the utterance and the output is one or more dialog acts (some utterances can have more than one dialog acts). For slot type prediction, the input is each predicted slot value together with the utterance, and the output is one of the predefined slot types. Given the limited numbers of various dialog-acts and slot types for classification, a standard CNN model is expected to achieve good performance.

[1] https://keras.io

Training size / OOV ratio		5% (16%)	10% (12%)	15% (4%)	20% (2%)
CNN	P	91.6	93.0	92.7	93.4
	R	61.7	62.5	65.8	69.2
	F	73.7	74.8	77.0	79.5
Seq2Seq w/ attention	P	81.3	83.6	84.1	85.3
	R	69.6	74.7	74.9	76.5
	F	75.0	78.9	79.2	80.7
Our model	P	86.9	86.4	85.7	85.9
	R	73.2	75.3	77.0	77.4
	F	**79.5**	**80.5**	**81.1**	**81.4**

Table 2: Results of slot value prediction with varying training size and OOV ratio.

Note that we can adopt other SLU frameworks as well (e.g., some joint frameworks), but given our focus in this work is to explore the hybrid Seq2Seq solutions for slot filling, we do not explore much on the SLU architecture, nor do we use any extra information (e.g., dialogue context). Despite the simplicity of our SLU system, it outperforms the prior state-of-the-art. In the whole process, neither manually designed features nor domain-specific rules are employed.

4.3 Baselines

We compare the overall SLU performance of our system against two existing baselines on DSTC2. One baseline (Williams, 2014) uses binary SVM classifiers to predict the existence of each slot-value pair and dialog act. The other (Barahona et al., 2016) uses CNN and LSTM jointly for classification.

For slot value prediction, since it is a sub-task of SLU and not reported in the prior work, we implemented another two models for it. One adopts CNN to classify an utterance into one or more slot values. The other uses the basic Seq2Seq attentional model (without Ptr-Net). Note that when learning this basic model, the target vocabulary is set to contain all the slot values in the training set.

4.4 Results of Slot Value Prediction

We first report the results on slot value prediction only. We compare the results of our proposed model and our own implemented baselines in Table 1, using precision, recall and F1.

We can see that the proposed hybrid model achieves the best F1 score. Although CNN has a high precision, it suffers from the low recall. By looking into the results for each slot type, it is ob-

Model	P	R	F
SLU1 (Williams, 2014)	84.6	76.2	80.2
SLU2 (Williams, 2014)	87.0	77.7	82.1
CNN+LSTM_w4 (Barahona et al., 2016)	-	-	83.6
CNN	**93.5**	78.5	85.3
Seq2Seq w/ attention	87.5	82.7	85.0
Our model	89.0	**82.8**	**85.8**

Table 3: Overall SLU performance.

Training Size		5%	10%	15%	20%
CNN	P	91.6	92.0	92.3	93.0
	R	67.5	70.4	71.7	72.7
	F	77.8	79.8	80.7	81.6
Seq2Seq w/ attention	P	82.8	87.2	86.4	87.9
	R	74.3	75.1	78.0	78.4
	F	78.3	80.7	82.0	82.9
Our model	P	84.9	86.3	88.4	88.0
	R	76.8	77.9	79.0	79.9
	F	**80.6**	**81.9**	**83.4**	**83.8**

Table 4: SLU results with varying training size.

served that CNN performs much poorer on non-enumerable types of slots such as *"food"* due to its high cardinality. While both our model and the basic Seq2Seq model have higher recall.

Since our assumption is that the proposed model can better handle the OOV problem, we analyze the OOV rate in the corpus to obtain more insight. By checking the percentage of slot values in the testing set that do not exist in the training, we find that the OOV problem in DSTC2 is not that severe, with a OOV ratio less than 0.1%. This could be a reason why our model does not obtain larger gain on the complete dataset. We therefore design more experiments in the next section to assess the model when the OOV problem is more severe.

4.5 OOV Slot Prediction

We create specific datasets by re-sampling from the original corpus. In particular, let group A denote all the training utterances that contain non-enumerable slots, and group B denote the rest of the training utterances. We randomly select 5%, 10%, 15%, and 20% of group A, plus the whole set of group B. In this way, we can create training data with less non-enumerable slot values thus resulting in a higher OOV ratio. The testing set is same as before. We compare the proposed model with the baselines on these four specific datasets with different OOV ratios (Table 2).

Input: **danish** food in the **centre** of town Output: danish centre | spanish centre | centre
Input: i would like **singaporean** food Output: singaporean | korean | None
Input: what about chiquito (**portuguese**) Output: chiquito | portuguese | None
Input: an **expensive** restaurant serving **cantonese** food Output: cantonese | portuguese expensive | expensive

Table 5: Examples of predicted slot values. Output is from the proposed model, Seq2Seq w/ attn, and CNN respectively (split by "|"). **Bold** denotes gold standard and "None" denotes empty result.

As shown in each column, on all the specific datasets, our model achieves the best performance. The CNN model performs much poorer than before in terms of the recall. We can see that by reducing the training size, the OOV ratio (indicated in the first row in the brackets) goes up, and the performance of all models decreases in general. While CNN and the basic Seq2Seq model decline 10.3% and 9.2% in F1 respectively using the smallest training set compared to using the complete one, our model is the most stable one with the least performance drop of 5.4%. The gain of our model over the other two becomes more significant with the larger OOV rate. This shows the capability of the Ptr-Net to correctly predict the OOV slots.

Overall, the results in Section 4.4 and 4.5 demonstrate the effectiveness of the proposed hybrid model for slot value prediction, especially when the training set is small and the OOV ratio is large.

4.6 SLU Results

Table 3 compares the results of the overall SLU task by our systems (incorporated with different slot value prediction models) and prior arts. All our systems outperform the prior work, and among them the one with the proposed hybrid model achieves the best F1 score.

We also conduct the similar OOV experiments as in Section 4.5 for SLU (Table 4). Similar trend is observed as discussed before. The performance of the proposed model with 20% training data already reaches that of the best system reported in the literature with 100% training data.

4.7 Case Study and Error Analysis

Table 5 gives some examples of slot values predicted by the proposed model and baselines. We

can see that for the less frequent slots, our model can still predict the values correctly, while without the Ptr-Net, the basic Seq2Seq model tends to generate words not appearing in the input, and CNN outputs nothing in many cases, which aligns with our assumption. We analyze the cases where Ptr-Net does not perform well and find several major types of errors: 1) partial prediction (e.g., detect only *"oriental"* for *"asian oriental food"*; 2) the prediction contains repetition of correct values; 3) speech recognition error although the prediction is proper if we look at the hypothesis itself (the third example). There are also cases where all the models fail to give a completely correct prediction, yet with different behaviors (the last example).

5 Conclusion

We adopt an attentional Seq2Seq model with Ptr-Net to predict slot values on dialogue data when only sentence-level semantic annotations are available. By switching between copying and generating words, this solution can bypass the need of word-level annotations and overcome the OOV issue which is very common in real-world spoken dialogue applications. It does not require any domain specific rules or dictionaries, and therefore can be easily adapted to new domains. Our model has achieved the state-of-the-art performance for both slot value prediction and SLU on the benchmark even with less training data.

Acknowledgments

We would like to thank Yifan He for helpful discussions and proofreading, and the anonymous reviewers for their valuable feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Lina M Rojas Barahona, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve Young. 2016. Exploiting sentence and context representations in deep neural models for spoken language understanding. In *Proceedings of COLING 2016*, pages 258–267.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 199–208.

Charles T Hemphill, John J Godfrey, George R Doddington, et al. 1990. The atis spoken language systems pilot corpus. In *Proceedings of the DARPA speech and natural language workshop*, pages 96–101.

Matthew Henderson, Milica Gašić, Blaise Thomson, Pirros Tsiakoulis, Kai Yu, and Steve Young. 2012. Discriminative spoken language understanding using word confusion networks. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 176–181. IEEE.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL Conference*, pages 263–272.

Matthew S Henderson. 2015. *Discriminative methods for statistical spoken dialogue systems*. Ph.D. thesis, University of Cambridge.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-LSTM and answer pointer. *arXiv preprint arXiv:1608.07905*.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *SIGDIAL Conference*, pages 282–291.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *AAAI*, pages 3365–3371.

Deyu Zhou and Yulan He. 2011. Learning conditional random fields from unaligned data for natural language understanding. *Advances in Information Retrieval*, pages 283–288.