Data Augmentation by Data Noising for Open-vocabulary Slots in Spoken Language Understanding

Hwa-Yeon Kim[1,2], Yoon-Hyung Roh[2], Young-Kil Kim[1,2]
University of Science and Technology[1]
Electronics and Telecommunications Research Institute[2]
hy.kim.aai@gmail.com, {yhroh, kimyk}@etri.re.kr

Abstract

One of the main challenges in Spoken Language Understanding (SLU) is dealing with 'open-vocabulary' slots. Recently, SLU models based on neural network were proposed, but it is still difficult to recognize the slots of unknown words or 'open-vocabulary' slots because of the high cost of creating a manually tagged SLU dataset. This paper proposes data noising, which reflects the characteristics of the 'open-vocabulary' slots, for data augmentation. We applied it to an attention based bi-directional recurrent neural network (Liu and Lane, 2016) and experimented with three datasets: Airline Travel Information System (ATIS), Snips, and MIT-Restaurant. We achieved performance improvements of up to 0.57% and 3.25 in intent prediction (accuracy) and slot filling (f1-score), respectively. Our method is advantageous because it does not require additional memory and it can be applied simultaneously with the training process of the model.

1 Introduction

Dialog processing enables dialogue between humans and voice assistants such as 'Siri' and 'Alexa'. In dialogue processing, spoken language understanding (SLU) is aimed at understanding and generating the user intention from an utterance. The user intention consists of an intent and slots, which are semantic entities, and it is generally defined variously according to the domain.

Neural networks (NNs) have been actively studied and applied to SLU. Xu and Sarikaya (2013) and Vu (2016) proposed models for SLU using Convolutional Neural Networks and many other researchers used recurrent neural networks (RNNs). Liu and Lane (2016) used bi-directional RNN models and applied the attention mechanism. They showed good results by using a joint learning method for both slot filling and intent prediction tasks. For considering the relationship between the two tasks, Wang et al. (2018) constructed two models for each task and Goo et al. (2018) used a 'slot-gate'.

Training an SLU model using an NN requires a large amount of training data labeled with slots and intents, which is expensive to build. In particular, plenty of corpora or dictionaries are required to recognize the value of an 'open-vocabulary' slot, such as a song title. In addition, it is difficult to predict the slot type of words used in the 'open-vocabulary' slot because there is neither a semantic restriction nor a length limit.

Kim et al. (2018) presented the features and examples of 'open-vocabulary' slot and proposed a new model that could effectively predict this type of slot. They exploited a long-term aware attention structure and positional encoding with multi-task learning of a character-based language model and intent detection model to focus more on relatively global information within a sentence.

The objective is to recognize slots, including 'open-vocabulary' slots, and predict the intent effectively by data augmentation. We propose a data noising method that reflects the characteristics of the 'open-vocabulary' slots. This method is advantageous in that it does not require additional memory. Moreover, it is performed simultaneously with the training of the model.

2 Related works

2.1 Data noising as a form of data augmentation

Data augmentation is a technique to avoid overfitting by increasing the size of the training datasets. This technique is widely used in many machine learning tasks. A typical method in natural language processing (NLP) is generating a sentence/corpus by replacing its words with their

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 97–102
Minneapolis, Minnesota, June 3 - 5, 2019. ©2017 Association for Computational Linguistics

synonyms based on rules, dictionaries, or ontology constructed by a person. In recent years, Kobayashi (2018) proposed a method of modifying sentences by the analogy of words to be replaced using an NN-based language model to achieve data augmentation.

One of the methods of data augmentation is data noising. Data noising is an effective technique for normalizing a neural network, and has been widely applied in fields such as computer vision and speech recognition. Its application in the field of NLP is relatively limited because NLP is based on discrete values like words and the quality of the data generated is not guaranteed. Several studies have used data noising for data augmentation. Iyyer et al. (2015); Bowman et al. (2015); Kumar et al. (2016) used a method for randomly dropping input word embedding. Xie et al. (2017) improved the performance by replacing a word with another word sampled from the unigram distribution or by blanking out as interpolation in language modeling. Cheng et al. (2018) improved the robustness of neural machine translation models against noisy inputs by randomly adding Gaussian noise to the word embedding and maintaining consistent behavior of the encoder to normal and perturbed inputs through adversarial learning. They showed that data noising is effective in normalizing sequence models based on neural networks.

2.2 Data augmentation for SLU

The NN model for SLU requires numerous labeled training corpora; therefore, some studies have attempted to improve SLU performance by data augmentation. Kurata et al. (2016) proposed a method of using encoder-decoder long short-term memory (LSTM) to generate labeled data. Hou et al. (2018) generated diverse utterances of existing training data through a data augmentation framework based on sequence-to-sequence generation. Yoo et al. (2018) proposed a data augmentation method that sampled similar words using a variational autoencoder. These methods have shown good performance improvements with data augmentation, but they require new models or consume additional memory.

3 Proposed Method

3.1 Motivation

First, we would like to cite an example of an utterance in the Snips dataset as the motive for the proposed method. Consider the sentence ‘i d like a table for midday at the unseen bean’; ‘the unseen bean’ is the value of the ‘restaurant_name’ slot. Even when people read ‘the unseen bean,’ it is difficult to recognize it as the name of a restaurant without prior knowledge. However, people can infer the slot type from the context (i.e., ‘i d like a table at’).

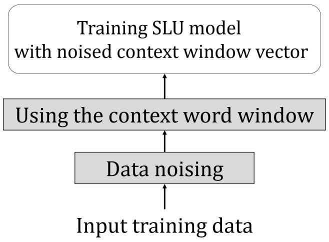
Training SLU model
with noised context window vector
Using the context word window
Data noising
Input training data

Figure 1: Proposed data augmentation method

‘Open-vocabulary’ slots, such as the name of a restaurant or the title of a song, have no restriction on the length or the specific patterns of content in the slot. Therefore, it is hard to recognize the slot type from only the words in the slots, but it is possible to predict them based on the surrounding words or the context.

Considering these linguistic features, the purpose of this study is to augment the data by transforming them into utterances with the same context/surrounding words but various slot values. It has the effect of creating utterance patterns.

3.2 Data noising for SLU

The flow of our proposed method is shown in Figure 1. It consists of two steps: data noising and using the context word window. When the training data are input, they become noised embedding vectors after data noising. Then, we use a context word window as the input to a layer of the neural network. These steps are performed per batch and the model trains with different noised data in each step. Because data augmentation is performed in the same embedding space, it does not need additional memory and is included in the training process.

Data noising: We propose a data noising method for SLU, which augments the training data by replacing the slot values with a random value while maintaining the surrounding context. The augmented data are used to train the NN for an utterance pattern. This method is shown in Figure 2 and follows the procedure described below.

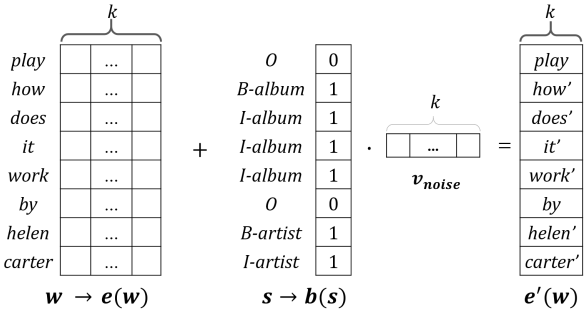

Figure 2: An example of data noising

1) Express an utterance $w = \{w_1, ..., w_T\}$ as an embedding vector $e(w) \in \mathbb{R}^{T\times k}$. T is the length of the word sequence and k is the size of the word embedding vector.

2) Extract the binary vector $b(s) = \{b_0, ..., b_T\}$ for the slot sequence $s = \{s_0, ..., s_T\}$ in the utterance w. The value of b_i is set to 1 when the i-th word w_i is the slot value (e.g., s_i is 'B-album'), and to 0 when it is not the slot value (e.g., s_i is 'O').

3) Multiply the binary vector $b(s)$ by the randomly sampled vector $v_{noise} \in \mathbb{R}^k$. Next, the noise is added only to the slot values, as shown in equation (1). This is to maintain unchanged the surrounding context of the slot value.

$$e'(w) = e(w) + b(s) \cdot v_{noise}, e'(w) \in \mathbb{R}^{T\times k} \qquad (1)$$

The noised embedding vector $e'(w_i)$ is located somewhere in the same embedding space as w_i. Because 'open-vocabulary' slots can contain *any* word, we do not need to know the words that are replaced. Therefore, we augment the training data by replacing the words in the 'open-vocabulary' slots with the embedding vector of *any* unknown word, through data noising.

Noised context word window: Mesnil et al. (2015) and Zhang and Wang (2016) used a context word window to improve the performance of the RNNs in slot filling. This can reflect the context information well by examining the surrounding words together. We also use the d-context word window as an input to the recurrent layer to reflect the contextual information. In this study, the noised context word window c_i^d is the result of the concatenation of the noised embedding vector $e'^k(w_i)$ of the center word w_i and the noised embedding vectors of the d previous words and d next words, as shown in equation (2).

	ATIS	Snips	MR
Train set	4,978	13,084	6,894
Evaluation set	-	700	766
Test set	893	700	1,521
Max. slot length	5	20	10

Table 1: Statistics of ATIS, Snips, and MR datasets.

$$c_i^d = [e'^k(w_{i-d}), .., e'^k(w_i), .., e'^k(w_{i+d})] \qquad (2)$$

4 Experiments and Results

4.1 Data

We experimented with three datasets: Airline Travel Information System (ATIS) (Hemphill et al., 1990), Snips[1], and MIT-restaurant (MR)[2]. The statistics of each dataset are shown in Table 1. We calculated the length of each slot and identified the maximum slot length. This was to numerically confirm whether each dataset had the characteristic of an 'open-vocabulary' slot, namely, there was no limit on the length of the slot value.

- **ATIS**: It is used in many SLU researches (Liu and Lane, 2016; Wang et al., 2018; Kim et al., 2018), including those on utterances for flight reservations. The training set is from ATIS-2 and ATIS-3 corpora, and the testing set is from ATIS-3, NOV93, and DEC94 datasets.

- **Snips**: It is an open-sourced NLU dataset of custom-intent-engines by Snips. It is used in SLU studies (Goo et al., 2018; Yoo et al., 2018). The Snips dataset contains user utterances from various domains, such as playing music or searching a movie schedule. It has 'open-vocabulary' slots, such as movie title.

- **MR**: It is a single-domain dataset, which is associated with restaurant reservations. MR contains 'open-vocabulary' slots, such as restaurant names.

4.2 Baseline and details of experiments

We set the attention based bi-directional LSTM model (Liu and Lane, 2016) without the label dependency as the baseline for the experiment. We

[1] https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines

[2] https://groups.csail.mit.edu/sls/downloads/restaurant/

applied our data augmentation method to the baseline and evaluated the following two cases: *Just add noise* (+Noise) , *Add noise and use context window* (+Noise, cw).

We followed the set-up in Liu and Lane (2016). We set the number of LSTM cells to 128, the batch size was 16, and the dropout rate was 0.5. We considered one layer and used the Adam optimizer (Kingma and Ba, 2015) for parameter optimization. The word embeddings were randomly initialized and then fine-tuned and their size was 128 for experiments with ATIS and MR, and 64 for experiments with Snips, for comparison with previous studies.

The noise vector was created with probability p and it defined the number of augmented data. In this paper, because we performed the data augmentation in batches during the training, the number of augmented utterances was defined as $(the\_number\_of\_step \times batch\_size) \times p$. The probability was set to 0.25, 0.5, 0.75, or 1.0. The noise vector was sampled randomly from the normal distribution or uniform distribution and its size was equal to the word embedding size. We set the mean of the normal distribution to 0.0, and the σ value to 0.1, 0.2, or 0.3. The range of the uniform distribution was set to [-0.2, 0.2] or [-0.5, 0.5].

As in previous SLU studies, we used the F1-score and the accuracy to evaluate the performance of the slot filling (SF) and the intent prediction(IP), respectively.

4.3 Performances of intent prediction and slot filling

Table 2 shows the performance improvements achieved by the proposed method versus the baseline. The proposed method showed clear improvements for the Snips and MR datasets. It is considered that the proposed method is effective in the two datasets because they have larger length of slots (as shown in Table 1) and more 'open-vocabulary' slots. Experimental results show that our approach improves slot filling of an unknown word or 'open-vocabulary' slots by learning the patterns of utterances. Examples illustrating the results can be found in Table 4. They show that the proposed method improves the slot filling of unknown words and 'open-vocabulary' slots by learning the utterance patterns.

	ATIS		Snips		MR
Method	Intent	Slot	Intent	Slot	Slot
Baseline	98.10	95.88	97.86	89.68	72.56
+Noise	98.32	95.80	98.57	92.58	74.60
+Noise, cw	**98.43**	**96.20**	**98.43**	**92.93**	**75.21**

Table 2: Performance of the proposed method with ATIS, Snips, and MR datasets.

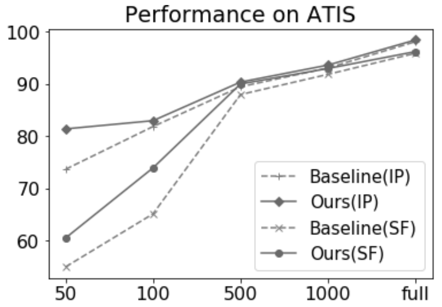

Figure 3: Performance of the proposed method with ATIS data of different sizes. The X-axis of each graph is the size of the training data. The Y-axis is the Accuracy (%) for Intent Prediction (IP) and F1-score for Slot Filling (SF).

Dataset	Model	Intent Prediction	Slot Filling
ATIS	Liu and Lane (2016)	98.21	95.98
	Kim et al. (2018)	98.54	95.93
	Wang et al. (2018)	**98.99**	**96.89**
	Ours	98.43	96.20
Snips	Goo et al. (2018)	97.00	88.80
	Yoo et al. (2018)	97.30	89.30
	Ours	**98.43**	**92.93**
MR	Yoo et al. (2018)	-	73.00
	Ours	-	**75.21**

Table 3: Comparison of the results for each dataset.

Additionally, the proposed method is more effective when the size of the training data is small. Figure 3 shows the performance according to the training data size with ATIS.

4.4 Comparison with previous studies

Table 3 shows the comparison of the performance of previous studies with each data set. In the case of the ATIS dataset, our method shows better performance than the 'Baseline' study (Liu and Lane, 2016) and a study targeting 'open-vocabulary' slots (Kim et al., 2018), but it is not state-of-the-art. However, we achieve the best performance among studies using the Snips and MR datasets.

Input	what s the weather in low *moor*
Baseline Pred.	O O O O O O I-timeRange
Proposed Pred.	O O O O O **B-city I-city**
Input	what is the *niceville* forecast in fm
Baseline Pred.	O O O O O O B-state
Proposed Pred.	O O O **B-city** O O B-state
Input	how can i view the show *corpus:* a home movie about selena
Baseline Pred.	O O O O O B-obj_type O O I-obj_nm B-obj_type I-obj_nm B-obj_nm
Proposed Pred.	O O O O O B-obj_type **B-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm I-obj_nm**
Input	add the song don t *drink* the water to my playlist
Baseline Pred.	O O B-music_item B-artist I-entity_nm I-entity_nm I-playlist I-playlist O B-playlist_owner O
Proposed Pred.	O O B-music_item **B-playlist I-playlist I-playlist I-playlist I-playlist** O B-playlist_owner O
Input	book a restaurant close by my daughters s work location with burrito three years from now
Baseline Pred.	O O B-rest_type B-spatial_relation I-spatial_relation O O I-poi O I-poi O B-served_dish B-timeR I-timeR I-timeR I-timeR
Proposed Pred.	O O B-rest_type B-spatial_relation I-spatial_relation **B-poi I-poi I-poi I-poi I-poi** O B-served_dish B-timeR I-timeR I-timeR I-timeR

Unknown word, Slot filling error, **Correct slot filling (Ground Truth)**

Abbreviation 'object': 'obj', '_name' : '_nm', 'restaurant' : 'rest', 'timeRange' : 'timeR'

Table 4: Examples of slot filling with the Snips dataset. 'Input' is the input utterance, and 'Baseline Pred.' is the slot filling result of the baseline model. 'Proposed pred.' is the result of applying the proposed method and is the ground truth. Unknown words are represented as italicized text. The slot filling errors are marked with underline and instances of correct slot filling are represented in bold text.

5 Conclusion

This paper focuses on data augmentation by reflecting the characteristics of 'open-vocabulary' slot in order to achieve better SLU. The experiments show that the proposed method outperforms the baseline, especially with datasets that include more 'open-vocabulary' slots. The proposed method has the following three advantages. Data augmentation can be performed during training. It does not need additional memory because it utilizes the input embedding space. It is straightforward and intuitive; hence, it can be easily applied to any model. In this paper, we added noise in all slots without classifying types of slots, i.e., without determining whether they were open-vocabulary slots. In our future work, we will perform additional experiments by classifying the types of slots and adding noise depending on the types. In addition, we plan to apply this method to Named Entity Recognition because it also has the problem of 'open-vocabulary' tags.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2019-0-0004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners).

References

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 10–21.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1756–1766.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, and Chih-Li Huo. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 753–757.

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

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, page 1234–1245.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1681–1691.

Jun-Seong Kim, Junghoe Kim, SeungUn Park, Kwangyong Lee, and Yoonju Lee. 2018. Modeling with recurrent neural networks for open vocabulary slots. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 2778–2790.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 452–457.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1378–1387.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *Proceedings of INTERSPEECH 2016*, pages 725–729.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Proceedings of INTERSPEECH 2016*, pages 685–689.

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*, pages 530–539.

Ngoc Thang Vu. 2016. Sequential convolutional neural networks for slot filling in spoken language understanding. In *Proceedings of INTERSPEECH 2016*, pages 3250–3254.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 309–314.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 78–83.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2018. Data augmentation for spoken language understanding via joint variational generation. In *arXiv preprint arXiv:1809.02305*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2993–2999.