DCEN: A Decoupled Context Enhanced Network For Few-shot Slot Tagging

[1]$YouliangYuan^{*}$,[1] $JiaxinPan^{*}$,[1] $XuJia$,[2] $LuchenLiu$,[1] $MinPeng^{\ddagger}$
[1]*School of Computer Science, Wuhan University*
Hubei, China
{ 2020282110194, pengm, pjx_1997, jia_xu }@whu.edu.cn
[2]*Institute of Information Engineering, Chinese Academy of Sciences)*
Beijing, China
liuluchen@iie.ac.cn

***Abstract*—Few-shot slot tagging is an important task in developing dialogue system. Most previous few-shot slot tagging models classify an item according to its similarity to the representation of each class. These models leverage context information implicitly through each words' contextual embedding. However, the entangled language features of words may interfere with context information, misleading the utilization of crucial slot features in few-shot scenario. To tackle these problems, we propose the Decoupled Context Enhanced Network (DCEN) for few-shot slot tagging. Different from previous models, we extract decoupled context explicitly to make full use of slot features contained in the context. Decoupled context includes two parts, local and global decoupled context information. We introduce a local extractor to extract local decoupled context by integrating information from adjacent words, and a global extractor based on transformer to extract global decoupled information by orthogonalization. Experimental results on SNIPS show that our model achieves the state-of-the-art performance with considerable improvements.**

I. Introduction

Slot tagging is a core task in natural language understanding (NLU), which is a key component of task-oriented dialogue system. Slot tagging is usually formulated as a sequence labeling problem [1], [2].

For slot tagging task, conditional random fields (CRF) [3] is a commonly used approach. Recently, neural models have become the de-facto standard for high-performance system. Although deep learning models have demonstrated their power for slot tagging task [4], [5], these models require abundant labeled training data in the target domain. To address data scarcity in new domain, few-shot learning technique [6], [7] becomes appealing. This technique extracts transferable knowledge among old domains and quickly transfers the model to a new domain with only a few examples [8]–[10].

The similarity-based few-shot learning methods classify an item in a new domain according to its similarity to the representation of each class, and have been widely used in classification problems [9], [11], [12]. To extract features of items in source domains, a general encoder is learned in prior rich-resource domains. These models obtain the representation of each class from few labeled samples of support set. Most few-shot slot tagging models [8], [10], [13] usually consider the following information: contextual word embedding, label semantic features and label dependency.

However, previous few-shot models do not consider the relationship between words and their corresponding contexts [8], [10]. Traditional contextual word embedding contains only a portion of the context information and loses the other word-independent context information. Also, in some cases, the information of the word itself may even lead to a misleading effect. For example, in figure 1, they will compute the similarity between the embedding of *Mojito* and the class representation of class *B-song* (average of the embeddings of *Angel* and *baby*). We classify the song by considering more the similarity of its context (*add* and *to*) because *Mojito* may be far away from *baby* (or *Angel*) in embedding space but the contexts of them are very similar. In this case, information of *Mojito* in contextual word embedding will be misleading when we classify.

In this paper, we propose a Decoupled Context Enhanced Network (**DCEN**) to explicitly utilize decoupled context information which does not contain information about the corresponding word to avoid misleading effect and the loss of context information. Decoupled context information includes two parts: local decoupled context information and global decoupled context information. The former is used to model the short-range context information, e.g., *B-song* usually come after *add*. The latter is used to model the long-range context information, e.g., Words between *my* and *playlist* are more likely to be *B-playlist* or *I-playlist* (The name of a playlist may have several words). We introduce Context Attention Module (**CAM**) which is based on self-attention to capture global decoupled context. For local context, we introduce Context Window Module (**CWM**). CWM obtains local decoupled context by obtaining collective representations of the adjacent words.

Our contributions are summarized as follows:

- We propose DCEN for few-shot slot tagging, which uses decoupled context information to avoid misleading effects caused by the representation of the words, and captures slot features of the context comprehensively.
- We explore approaches to make use of local and global

‡ Corresponding author.
∗ Equal contribution.

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533361

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

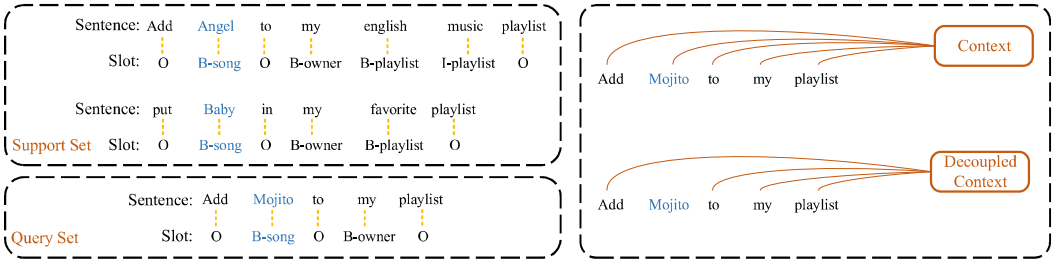

Fig. 1. Left part is an example of few-shot slot tagging. Right part shows the context information and decoupled context information of the blue words. Decoupled context is different for each word and no information about the word is in the corresponding decoupled context.

decoupled context information. Extensive experiments show separating word and context features are both effective for improving few-shot slot tagging performance.

- We demonstrate the effectiveness of our model on the benchmark dataset SNIPS and achieve SOTA results. Further analyses show that by fusing the extracted decoupled context, our model can release underutilization of limited context information.

II. Related Work

A. Slot Tagging

For slot tagging task, conditional random fields (CRF) [3] and recurrent neural networks (RNN) [14], [15] are commonly used methods. Recently, joint models [16], [17] become popular because they are able to consider the correlated relationship between slot tagging and intent detection. [18]–[20] propose model to improve performance of slot tagging and intent detection via mutual interaction.

B. Few-shot

Traditional methods for the few-shot learning in image classification field primarily focus on metric learning [7], [21], which aims to learn a general distance metric and use it in the new domain. [12] further develops Prototypical Network by constructing a task-adaptive space based on label references. These models classify an item according to its similarity to each class's representation.

In natural language processing, few-shot researchers' focus is primarily on text classification [9], [22], [23]. Also, zero-shot intent detection has been explored [24], [25] with a Gaussian mixture model and Capsule Neural Network. Recently, researchers are paying more and more attention to the few-shot learning of the slot tagging task. [26] investigates few-shot slot tagging using additional regular expressions. Zero-shot slot tagging approaches [27]–[29] achieve impressive performance by using label name semantic features (description of the class label and examples of slot values) in zero-shot setting. [8] explores few-shot named entity recognition (NER) with Prototypical Network. [10] exploits the Label-enhanced TapNet with collapsed dependency transfer (CDT) for both slot tagging and NER tasks. By considering both label dependency transferring and label name semantics, [10] achieves much better performance.

III. Approach

In this section, we describe the proposed DCEN model in details (shown in Figure 2). The model first gets the embeddings of each word from BERT [30]. Then L-TapNet is used to calculate the word emission score in the word part. To extract decoupled context of each word, Context Attention Module and Local Window Module are utilized in context part. The Context Attention Module attends full-range contextual information by modified self-attention while the Local Window Module focuses on short-range contextual information by gathering features from neighbouring words. Finally, we combine context emission score with word emission score as the emission score of CRF.

A. Word Part

In word part, we feed the word embedding to L-TapNet. After that, the word emission score is calculated by vector projection similarity function with output of L-TapNet.

L-Tapnet TapNet is a few-shot image classification model. Different from previous few-shot model (such as Prototypical Network), TapNet calculate word-label similarity in a projected embedding space, where the words of different labels are well-separated. L-TapNet further develops TapNet by constructing a projection space with label semantics.

Vector Projection We use vector projection similarity function (VPB) from [13] to compute similarity. The similarity is calculated by dot product between word embedding $\boldsymbol{x_i}$ and each normalized label vector $\boldsymbol{c_k}$. In order to reduce false positive errors, the half norm of each label vector is utilized as an adaptive bias term:

$$SIM(\boldsymbol{x_i}, \boldsymbol{c_k}) = \boldsymbol{x_i}^T \frac{\boldsymbol{c_k}}{\| \boldsymbol{c_k} \|} - \frac{1}{2} \| \boldsymbol{c_k} \|$$

This function can help eliminate the impact of $\boldsymbol{c_k}$'s norm which may be large enough to dominate the similarity metric.

B. Context Part

In context part, we feed the decoupled embedding to TapNet. Using Vector Projection, context emission score is obtained by combining local and global context emission scores. The TapNet and Vector Projection are the same as section III-A.

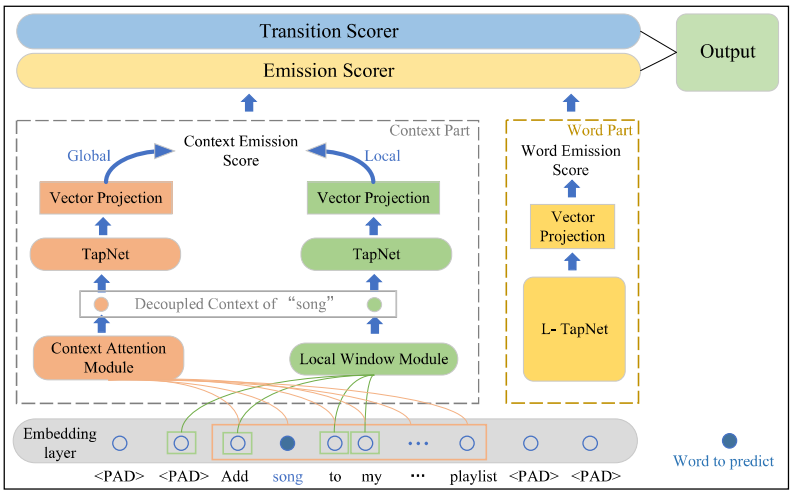

Fig. 2. Architecture of our proposed DCEN model. It consists of four parts: a) Embedding layer tranforms words to embedding. b) Word part calculates the word emission scores for the query instance based on the prototypes derived from the support set. c) Context part calculates the context emission scores. d) Transition scorer uses collapsed dependency transfer to compute transition score.

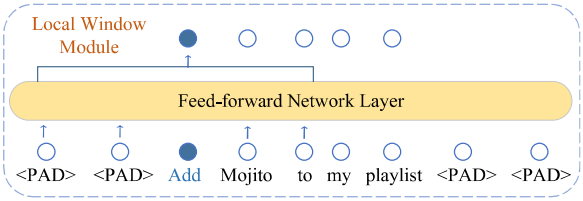

Fig. 3. Illustration for Context Window Module. We concatenate the embedding of adjacent words of the word (the blue one) and feed new embedding to Feed-forward Network.

Context Window Module Given an utterance $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ with n words, where x_i is the i^{th} word of the utterance. We expand $\boldsymbol{x}$ to $\boldsymbol{x'}$, in which pad_i is zero tensor:

$$\boldsymbol{x'} = (pad_1, \ldots, pad_w, x_1, \ldots, x_n, pad_1, \ldots, pad_w)$$

where w is window size (as shown in Figure 3) which determines the coverage of the input. In order to obtain decoupled context about $x_i^{'}$, we exclude $x_i^{'}$ from the input. For each word, we can obtain its local decoupled context embedding $\boldsymbol{e_i^l}$ by integrating information from adjacent words:

$$\boldsymbol{e_i} = [E(x_{i-w}^{'}); \ldots; E(x_{i-1}^{'}); E(x_{i+1}^{'}); \ldots; E(x_{i+w}^{'})]$$

$$\boldsymbol{e_i^l} = FFN(\boldsymbol{e_i})$$

$FFN(\cdot)$ is Feed Forward Network and $E(x_i^{'})$ is the embedding of $x_i^{'}$.

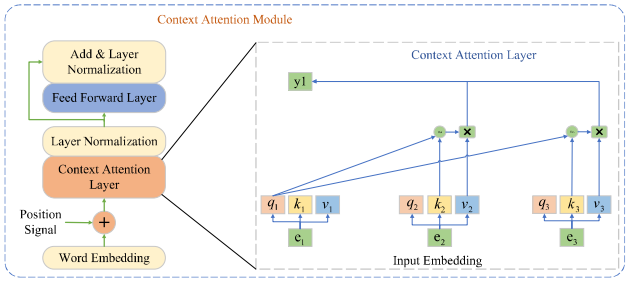

Fig. 4. The architecture of the global context extractor and illustration of the context attention layer.

Context Attention Module For an utterance $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ with n words, we first get $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)$ where $e_i = E(x_i)$. Then we add position signal [31] to $\boldsymbol{e}$ and obtain $\boldsymbol{e'} = (e'_1, e'_2, \ldots, e'_n)$. Afterwards, $\boldsymbol{e'}$ is fed to context attention layer (see Figure 4):

$$y_i' = \sum_{j=1}^{n} c_{ij} v_j$$

$$\boldsymbol{C} = softmax(\frac{\boldsymbol{QK}^T - diag[(\boldsymbol{Inf_{(1)}}, \ldots, \boldsymbol{Inf_{(n)}}]}{\sqrt{d_k}})$$

To better model the long-range context, we adopt transform encoder [31], which maps the matrix of input vectors to queries ($\boldsymbol{Q}$), keys ($\boldsymbol{K}$) and values ($\boldsymbol{V}$ and v_j is a element for the j^{th} word) matrices by using three different linear projections and $\boldsymbol{C}$ is a weight matrix (c_{ij} is a element). d_k denotes the dimension of keys, and $diag[\cdot]$ means diagonal matrix. $\boldsymbol{Inf}$ is positive infinity.

Vanilla Tranformer is designed to acquire precise word embeddings by considering global self-attention. However, our interest is to disentangle global decoupled context information of each word. To achieve this, we adopt mask technology as the decoupling method. By masking the innate information

of the word through orthogonalization, the obtained global context information circumvents misleading effects of each word's own meaning. After $softmax$ function, the element (c_{ii}) corresponding to the position of the negative infinity element will become zero. In this way, the output of context attention layer y_i' is able to neglect v_i (which means the information of the related word). After layer normalization, we send y_i' to the feed forward layer and obtain the embedding of global decoupled context.

Afterwards, similarity of context $SIM_{context}$ is computed as:

$$SIM_{context} = \alpha \cdot SIM_{local} + \beta \cdot SIM_{global}$$

where α and β are hyper-parameters. These two hyper-parameters determine the extent to which the model refers to local and global information. Both types of information are equally important for our approach, so we set both α and β to 0.5 (We have tried to change α and β, and found that the performance is not sensitive to the ratio of the two in the appropriate range. When the ratio of α and β is very large or small, it is equivalent to using only local extractor or global extractor).

Finally, emission score E is computed as:

$$E = p * SIM_{context} + (1 - p) * SIM_{word}$$

where p is a hyper-parameter. p determines the extent to which we consider decoupled context.

IV. Experiments

In our experiment, we evaluate our model following the dataset split provided by [10] on SNIPS [32]. Each episode contains support set $\mathcal{S}$ and query set $\mathcal{Q}$. There are 7 different domains in SNIPS for slot tagging: Weather (We), Music (Mu), PlayList (Pl), Book (Bo), Search Screen (Se), Restaurant (Re) and Creative Work (Cr).

Statistical analyses of the original datasets are provided in the table I.

TABLE I
STATISTICS OF THE ORIGINAL DATASET. THE NUMBER OF LABELS ("LABELS") IS COUNTED IN INSIDE/OUTSIDE/BEGINNING (IOB) SCHEMA.

Dataset	Domain	#Sent	#Labels
SNIPS	We	2100	17
	Mu	2100	18
	Pl	2042	10
	Bo	2056	12
	Se	2059	15
	Re	2073	28
	Cr	2054	5

[10] reorganizes the dataset for few-shot slot tagging in episode data setting. The overview of the data split on SNIPS is shown in the table II.

TABLE II
OVERVIEW OF FEW-SHOT SLOT TAGGING DATA FROM SNIPS. "$\boldsymbol{Avg.|S|}$" REFERS TO THE AVERAGE SUPPORT SET SIZE OF EACH DOMAIN, AND "**#SENT**" INDICATES THE NUMBER OF LABELLED SAMPLES IN BATCHES OF ALL EPISODES.

Domain	$\boldsymbol{Avg.\|S\|}$ (1-shot)	#Sent (1-shot)	$\boldsymbol{Avg.\|S\|}$ (5-shots)	#Sent (5-shots)
We	6.15	2000	28.91	1000
Mu	7.66	2000	34.43	1000
Pl	2.96	2000	13.84	1000
Bo	4.34	2000	19.83	1000
Se	4.29	2000	19.27	1000
Re	9.41	2000	41.58	1000
Cr	1.30	2000	5.28	1000

For each dataset, we utilize 5 domains for training, one domain for validation and one domain for evaluation. And we report the average F1 scores at the episode level as well. For each experiment, we run it ten times with different random seeds to alleviate the randomness in neural network training.

In our experiments, we use the uncased BERT-Base [30] to obtain original word embedding. The models are trained using ADAM [33] (batch size 4 and learning rate 1e-5). For local context extractor, we set window size as 2 during training and testing. We set coupling coefficients α as 0.5 and β as 0.5. We set the proportion of decoupled context information p as 0.5. In 5-shot setting, our batch size is 2. We train our models for 2 iterations in 1-shot setting and 4 iterations in 5-shot setting. Then, we save the parameters with best F1 scores on the validation domain. We run it ten times with ten different seed for each experiment to control randomness.

A. Baseline

Bi-LSTM is bidirectional LSTM [34] with Glove [35] embedding. It is trained on the support set and tested on the query sample.

SimBERT This model directly predicts labels according to cosine similarity of word embedding of frozen BERT.

TransferBERT is a domain transfer model with the NER setting of BERT. We pretrain it on source domains and fine-tune it on target domain support set (only transfer bottleneck feature). Learning rate is set as 1e-5 in training and fine-tuning.

Matching Network (MN) We employ the matching network [7] with BERT embedding for classification.

L-WPZ+CDT (LWPZC) WarmProtoZero (WPZ) [8] is a few-shot sequence labeling model which regards sequence labeling as the classification of each word. This model contains a Prototypical Network [21] which trained on source domains and directly utilizes it for word-level classification on the target domain. [10] enhanced WPZ by utilizing BERT and CDT. **"L-"** in **L-WPZ** means that label-enhanced prototypes are applied by using semantic of the label name.

L-TapNet L-TapNet [10] is a few-shot CRF model for slot tagging. It computes the label transition score with collapsed dependency transfer and computes the emission score with Label-enhanced TapNet. This model is the previous state-of-the-art method for few-shot tagging.

TABLE III
F1 SCORES ON FEW-SHOT TAGGING OF SNIPS.

	MODEL	We	Mu	Pl	Bo	Se	Re	Cr	Avg.
1-shot	Bi-LSTM	10.36±0.36	17.13±0.61	17.52±0.76	53.84±0.57	18.44±0.44	22.56±0.10	8.64±0.41	21.21±0.46
	SemBERT	36.10±0.00	37.08±0.00	35.11±0.00	68.09±0.00	41.61±0.00	42.82±0.00	23.91±0.00	40.67±0.00
	TransferBERT	55.82±2.75	38.01±1.74	45.65±2.02	31.63±5.32	21.96±3.98	41.79±3.81	38.53±7.42	39.06±3.86
	MN	21.74±4.60	10.68±1.07	39.71±1.81	58.15±0.68	21.21±1.20	32.88±0.64	69.66±1.68	36.72±1.67
	L-TapNet	73.21±1.46	60.97±2.23	69.24±3.34	84.53±1.23	74.44±3.54	72.48±0.98	67.44±2.30	71.76±2.15
	WPZC	73.56±0.93	58.40±1.11	68.93±0.95	82.32±0.78	**79.69**±0.55	73.40±0.75	70.25±1.22	72.37±0.90
	LWPZC	73.19±1.65	58.62±1.02	68.26±0.42	83.54±0.62	77.88±0.59	73.48±1.13	69.54±1.64	72.07±1.01
	w/o CAM (ours)	75.19±1.49	60.80±0.94	73.40±1.76	85.41±0.63	77.35±1.93	75.79±1.10	65.80±0.74	73.39±1.23
	w/o CWM (ours)	77.90±1.70	60.61±1.28	69.95±1.11	**86.29**±0.78	77.88±0.89	75.51±1.99	**71.10**±1.26	74.18±1.29
	DCEN (ours)	**78.26**±0.92	**61.95**±1.33	**73.55**±0.87	86.02±0.85	78.76±1.49	**76.71**±0.65	68.57±2.33	**74.83**±1.21
	MODEL	**We**	**Mu**	**Pl**	**Bo**	**Se**	**Re**	**Cr**	**Avg.**
5-shot	Bi-LSTM	25.17±0.42	39.80±0.52	46.13±0.42	74.60±0.21	53.47±0.45	40.35±0.52	25.10±0.94	43.52±0.50
	SemBERT	53.46±0.00	54.13±0.00	42.81±0.00	75.54±0.00	57.10±0.00	55.30±0.00	32.38±0.00	52.96±0.00
	TransferBERT	59.41±0.30	42.00±2.83	46.07±4.32	20.74±3.36	28.20±0.29	67.75±1.28	58.61±3.67	46.11±2.29
	MN	36.67±3.64	33.67±6.12	52.60±2.84	69.09±2.36	38.42±4.06	33.28±2.99	72.10±1.48	47.98±3.36
	L-TapNet	84.45±1.38	70.83±1.38	81.26±2.36	88.84±1.04	87.16±1.02	81.45±1.14	75.95±1.88	81.42±1.46
	WPZC	82.91±0.85	69.23±0.56	80.85±1.18	90.69±0.43	86.38±0.47	81.20±0.45	**76.75**±1.59	81.14±0.79
	LWPZC	82.93±0.59	69.62±0.46	80.86±1.04	**91.19**±0.37	86.58±0.63	**81.97**±0.57	76.02±1.65	81.31±0.76
	w/o CAM (ours)	85.58±0.42	72.14±1.03	**84.54**±0.75	89.06±0.49	**88.95**±0.81	81.76±1.02	68.76±1.88	81.54±0.92
	w/o CWM (ours)	85.27±0.89	71.08±1.89	78.21±0.81	90.54±0.62	85.71±1.31	81.10±0.52	75.53±2.85	81.06±1.27
	DCEN (ours)	**85.76**±1.30	**74.85**±1.37	82.67±1.42	90.28±0.72	88.58±0.45	81.66±0.74	72.55±1.64	**82.34**±1.09

We use results of Bi-LSTM, SemBERT, TransferBERT and MN from [10] and results of LWPZC and WPCZ from [13]. All of our models use vector projection (VPB) similarity function in the experiments. All experiments are performed on Pytorch and MindSpore [1].

B. Main Results

Table III shows the result in both 1-shot and 5-shot slot tagging of SNIPS. Each column shows the F1 scores of taking the domain of the column header as target domain and other domains as source domain (train and dev). **"w/o CAM/CWM"** means CAM/CWM is removed. Our method can outperform all baselines. As shown in the table, DCEN achieves the best performance and it outperforms L-TapNet by F1 scores of 3.07 and 0.92 on 1-shot and 5-shot experiment in average respectively. Compared to L-TapNet, our model has better performance in almost all target domains. We contribute this to our explicit consideration of the relationship between contexts and words (In L-TapNet, this relationship is only implicitly manifested in the transition module of CRF and pretrained embedding).

C. Analysis

By comparing L-TapNet with our models, we can find that our proposed Context Window Module and Context Attention Module can improve the performance of the similarity-based model in both 1-shot and 5-shot cases which demonstrates the effectiveness of our approach. This model uses two extractors to capture decoulped context information and further develop the previous similarity-based model by making full use of context information.

[1]https://www.mindspore.cn/

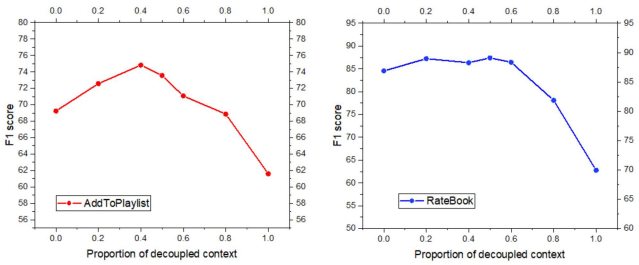

Fig. 5. Model performance at different proportion of decoupled context information. p is the proportion of decoupled context information (as mentioned before). When $p = 0$, the model degenerates to L-TapNet.

Information composition To further understand the effect of decoupled context information, we test our model on *AddToPlaylist* and *RateBook* at different proportion of decoupled context information in Fig 5. In the left figure, most slots in *AddToPlaylist* (such as *B/I-artist*, *B/I-entity_name*) are more relevant with their decoupled context. As p increases from 0 to 0.4, the F1 score improves significantly. This result proves the effectiveness of decoupled context information. On the contrary, the slots (such as *B/I-rating_unit*, *B/I-rating_value*) in *RateBook* of the right figure are less relevant with their decoupled context. In this case, the fact that the best performance is achieved when $p = 0.5$ rather than $p = 0$ shows the decoupled context can still improve the performance. Notwithstanding we only use decoupled context information ($p = 1$), our model has acceptable performance in *AddToPlaylist*. This result affirms that our model effectively exploits context information again.

It is worth to note that the best performance occurs when decoupled context information is used (when p is between 0.4 and 0.6). This illustrates the necessity of incorporating decoupled context with original word information.

Ablation Study Ablation study results are shown in

table IV. Respectively, each of our components is removed, including: CAM and CWM. The results in the table IV show that both CAM and CWM play important roles in our model which means both local and global decoupled context are beneficial for slot tagging. As shown in the table III, our model has better performance in some domains if we remove CAM or CWM. We speculate that this is due to the fact that global/local information is not important for some slots. The degree of dependence of different slot on local/global context is not the same and the distribution of slot is different in different domains. But from the final results, considering both local and global information is the best choice (best average performance and better robustness).

TABLE IV
ABLATION STUDY OVER DIFFERENT COMPONENTS ON SLOT TAGGING TASK. RESULTS ARE AVERAGED F1-SCORE OF ALL DOMAINS.

MODEL	1-shot	5-shots
Ours	74.83	82.34
-CAM	-1.44	-0.79
-CWM	-0.65	-1.27
-all	-3.07	-0.92

V. CONCLUSION

In this paper, we propose the Decoupled Context Enhanced Network for few-shot slot tagging, which considers word information and context information separately to alleviate misleading effects of innate meanings of words. To extract the decoupled context, we use Context Window Module as local context extractor and Context Attention Module as global context extractor. Experimental results validate both local and global context extractors are effective.

ACKNOWLEDGMENT

Thanks for the financial support of the National Key R&D Program of China under Grant No. 2018YFC1604003, General Program of Natural Science Foundation of China (NSFC) under Grant No. 61772382 and No. 62072346 and Key R&D Project of Hubei Province under Grant No. 2020BAA021. And thank for MindSpore, a new deep learning computing framework.

REFERENCES

[1] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 530–539, 2015. I

[2] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Interspeech 2016*, pp. 685–689, 2016. I

[3] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001. I, II-A

[4] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 2078–2087, Association for Computational Linguistics, Nov. 2019. I

[5] H. E, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5467–5471, Association for Computational Linguistics, July 2019. I

[6] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006. I

[7] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, pp. 3630–3638, 2016. I, II-B, IV-A

[8] A. Fritzler, V. Logacheva, and M. Kretov, "Few-shot classification in named entity recognition task," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 993–1000, 2019. I, I, II-B, IV-A

[9] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun, "Induction networks for few-shot text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 3904–3913, Association for Computational Linguistics, Nov. 2019. I, II-B

[10] Y. Hou, W. Che, Y. Lai, Z. Zhou, Y. Liu, H. Liu, and T. Liu, "Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 1381–1393, Association for Computational Linguistics, July 2020. I, I, II-B, II-B, IV, IV, IV-A, IV-A, IV-A

[11] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018. I

[12] S. W. Yoon, J. Seo, and J. Moon, "Tapnet: Neural network augmented with task-adaptive projection for few-shot learning," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 7115–7123, PMLR, 2019. I, II-B

[13] S. Zhu, R. Cao, L. Chen, and K. Yu, "Vector projection network for few-shot slot tagging in natural language understanding," *arXiv preprint arXiv:2009.09568*, 2020. I, III-A, IV-A

[14] 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*, pp. 78–83, IEEE, 2013. II-A

[15] 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)*, pp. 189–194, IEEE, 2014. II-A

[16] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *arXiv preprint arXiv:1609.01454*, 2016. II-A

[17] C. Li, L. Li, and J. Qi, "A self-attentive model with gate mechanism for spoken language understanding," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3824–3833, 2018. II-A

[18] Y. Wang, Y. Shen, and H. Jin, "A bi-model based rnn semantic frame parsing model for intent detection and slot filling," *arXiv preprint arXiv:1812.10235*, 2018. II-A

[19] P. Niu, Z. Chen, M. Song, *et al.*, "A novel bi-directional interrelated model for joint intent detection and slot filling," *arXiv preprint arXiv:1907.00390*, 2019. II-A

[20] L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, and T. Liu, "A co-interactive transformer for joint slot filling and intent detection," *arXiv preprint arXiv:2010.03880*, 2020. II-A

[21] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, pp. 4077–4087, 2017. II-B, IV-A

[22] L. Yan, Y. Zheng, and J. Cao, "Few-shot learning for short text classification," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29799–29810, 2018. II-B

[23] R. Geng, B. Li, Y. Li, J. Sun, and X. Zhu, "Dynamic memory induction networks for few-shot text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 1087–1094, Association for Computational Linguistics, July 2020. II-B

[24] G. Yan, L. Fan, Q. Li, H. Liu, X. Zhang, X.-M. Wu, and A. Y. Lam, "Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1050–1060, 2020. II-B

[25] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 3090–3099, Association for Computational Linguistics, Oct.-Nov. 2018. II-B

[26] B. Luo, Y. Feng, Z. Wang, S. Huang, R. Yan, and D. Zhao, "Marrying up regular expressions with neural networks: A case study for spoken language understanding," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 2083–2093, Association for Computational Linguistics, July 2018. II-B

[27] A. Bapna, G. Tür, D. Hakkani-Tür, and L. Heck, "Towards zero-shot frame semantic parsing for domain scaling," in *Proc. Interspeech 2017*, pp. 2476–2480, 2017. II-B

[28] S. Lee and R. Jha, "Zero-shot adaptive transfer for conversational language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6642–6649, 2019. II-B

[29] D. Shah, R. Gupta, A. Fayazi, and D. Hakkani-Tur, "Robust zero-shot cross-domain slot filling with example values," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 5484–5490, Association for Computational Linguistics, July 2019. II-B

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019. III, IV

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017. III-B, III-B

[32] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018. IV

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. IV

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. IV-A

[35] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014. IV-A