ELASTIC CRFS FOR OPEN-ONTOLOGY SLOT FILLING

Yinpei Dai[†]∗*, Yichi Zhang*[†]∗*, Hong Liu*[†]*, Zhijian Ou*[†]*, Yi Huang*[‡]*, Junlan Feng*[‡]

[†]Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China
[‡] China Mobile Research Institute

dyp16, zhangyic17@mails.tsinghua.edu.cn, ozj@tsinghua.edu.cn, fengjunlan@chinamobile.com

ABSTRACT

Slot filling is a crucial component in task-oriented dialog systems that is used to parse (user) utterances into semantic concepts called slots. An ontology is defined by the collection of slots and the values that each slot can take. The most widely used practice of treating slot filling as a sequence labeling task suffers from two main drawbacks. First, the ontology is usually pre-defined and fixed and therefore is not able to detect new labels for unseen slots. Second, the one-hot encoding of slot labels ignores the correlations between slots with similar semantics, which makes it difficult to share knowledge learned across different domains. To address these problems, we propose a new model called elastic conditional random field (eCRF), where each slot is represented by the embedding of its natural language description and modeled by a CRF layer. New slot values can be detected by eCRF whenever a language description is available for the slot. In our experiment, we show that eCRFs outperform existing models in both in-domain and cross-domain tasks, especially in predicting unseen slots and values.

***Index Terms*—** Open ontology, slot filling, conditional random fields, dialog systems

1. INTRODUCTION

Slot filling [1, 2] is a crucial component in task-oriented dialog systems and parses (user) utterances into semantic concepts in terms of a set of named entities called slots. The example in Figure 1 contains the slots `time` and `movie`. In parsing, some span in the utterance is identified as the slot value for some slot; e.g., here, *"6 pm"* is marked as the slot `time`. An ontology, which describes the scope of semantics that the dialog system can process, is defined by the collection of slots and the values that each slot can take. A widely used practice for slot filling is to introduce IOB tags [3] and assign a label to each token in the utterance. A label, e.g., `B-time`, is a combination of the slot name and one of the IOB tags.These labels are then used to identify the values for different slots from the utterance. In this manner, slot filling is treated as a sequence labeling task, as illustrated in Figure 1, for which the two dominant classes of methods are based on recurrent neural networks (RNNs) [1] and conditional random fields (CRFs) [4], respectively. This practice has been widely employed for slot filling [2, 5] and many other similar sequence labeling problems [6]. However, this practice suffers from two drawbacks.

Utterance	6	pm	for	the	movie	called	avatar
Slot labels	B-time	I-time	O	O	O	O	B-movie

Fig. 1. An example of slot filling in the movie domain.

First, currently, most slot-filling methods are unable to predict new labels for unseen slots. The ontology is usually pre-defined and fixed. It is difficult to accommodate new semantic concepts (slots) in slot filling. However, users may often add new semantic concepts in a domain and dialog systems are expected to work across an increasingly wide range of domains. Thus, it is highly desirable for slot-filling models to be able to handle new slots, whether in-domain or cross-domain, with the least expense being incurred after training on a certain domain. In this paper, we are interested in developing such open-ontology slot filling, which means that the collection of slots and values is open-ended for slot filling. Second, in current slot-filling models [5, 7], slot labels are generally encoded as one-hot vectors. However, slot labels are not merely discrete classes. There are natural language descriptions for each slot, e.g., the description "number of people" for the slot `#people`. This one-hot encoding ignores the semantic meanings and relations for slots, which are implicit in their natural language descriptions and useful for slot filling.

There are prior efforts to address the above two drawbacks. The difficulty of transferring between domains could be partly alleviated with multi-task learning [8, 9, 10], by performing joint learning on multiple domains. Practically, varying only the last output layer for different domains and sharing the parameters of the rest layers has shown to be a successful approach [11]. In this approach, the slot-filling model can leverage all available multi-domain data and transfer them to handle those slots with sparse training data. However, ba-

∗These authors contributed equally to this work. Supported by NSFC 61473168, Ministry of Education and China Mobile joint funding MCM20170301.

sically, this multi-task learning approach is unable to predict labels for zero-shot slots (namely those slots that are unseen in training data and whose values are unknown). It can be seen that this difficulty is also related to the drawback of one-hot encoding slot labels, which hinders the exploitation of semantic relations and shared statistical properties between different slots. A recent work [12] proposes utilizing slot label descriptions towards zero-shot slot filling by introducing slot encodings from natural language descriptions. Basically, they use RNN-based sequence labeling, taking the slot encoding vector as an additional conditional input and outputting the IOB tags in each position. Sequence labeling is carried out independently for all slots. Though yielding promising results, there are two shortcomings. First, independent sequence labeling may make conflicting predictions. Second, interactions between slots are ignored in sequence labeling.

CRFs have been shown to be one of the most successful approaches for sequence labeling, especially for capturing the interactions between labels. A widely used method is to implement a CRF layer on top of features generated by a RNN [1]. These recent neural CRFs are different from conventional CRFs, which mainly use discrete indicator features. However, these recent CRFs still work with a closed set of labels. In this paper, we propose a novel neural CRF model, called elastic CRF (eCRF), for open-set sequence labeling, by leveraging label descriptions inspired from [12]. The key idea of eCRFs is to use slot descriptions to create semantically meaningful IOB tags [3], which are further used for a new calculation of potential functions in the CRF framework. Compared to traditional fixed IOB tags in original CRFs, our eCRFs are able to process new slots unseen during training without retraining the model. Such flexibility is the motivation for calling it an "elastic" CRF model.

The eCRFs are powerful models for open-ontology slot filling. Intuitively, the node potentials of eCRFs combine the neural features of both the utterance and the slot descriptions, and the edge potentials model the interactions between different slots. In the experiments, we make use of the Google simulated dataset [13], and re-split the dataset according to the in-domain task and the cross-domain task, which focus on the challenge of handling unseen values and unseen slots, respectively. The results show that eCRFs significantly outperform not only a BiLSTM baseline but also the concept tagger (CT) in [12] for both tasks, especially in predictions of unseen slots and values.

In Section 2, we discuss related work. The new eCRF model is detailed in Section 3. Section 4 describes the dataset and task formulations. Section 5 presents the experiments, followed by the conclusion in Section 6.

2. RELATED WORK

One line of related work is zero-shot slot-filling learning [14, 15]. The term *open ontology* referred in this paper is a different name for zero-shot slot filling in spoken language understanding (SLU) for dialog systems. Zero-shot learning has been applied in various of SLU tasks. The authors of [16] leverage the intent embeddings to detect new intent labels which are not included in the training data. Additionally, [12] exploits the slot label descriptions to parse the novel semantic frames for domain scaling and [17] extends the natural language generation module to generalize the responses into an unseen domain via latent action matching. The authors of [18] propose utilizing both the slot description and a small number of examples of slot values to enhance model robustness. In [19], the authors focus on multi-turn zero-shot slot filling in conversation. These studies have utilized the natural language descriptions of the labels, and by constructing the semantic encoder to take the label descriptions as inputs, any new labels in the testing phrase can still be predicted by the model. Our eCRFs also use this semantic encoder structure. However, unlike processing each label description separately in [12], eCRFs are trained and tested by jointly exploiting all possible slot descriptions at one time. Thus, they could capture relations between slot labels and relieve the burden of adjusting the oversampling ratio.

Another line of related work is models for slot filling. CRFs have been extensively applied in traditional slot-filling tasks [20, 21, 22, 23, 24, 25, 26], but are restricted by a fixed set of labels. With the progress of deep learning, state-of-art slot-filling methods usually utilize BiLSTM networks [9, 27, 28, 29]. Extended models, such as encoder–decoder [5] and memory network [30] designs, are explored. More recently, [31] proposes a coarse-to-fine approach (Coach) for cross-domain slot filling, which detects the value span boundary first and then predicts the specific fine types for the slot entities. With the advance of pre-trained models [32], there are also many work [33, 34, 35, 36, 37] that adapt the well-studied machine reading comprehension (MRC) framework to solve open-ontology slot filling or using pre-tained dialogue models to generate slot labels [38, 39, 40, 41, 42]. Motivated by the BiLSTM-CRF architecture [43, 20, 44], our eCRFs combine the representation power of deep neural networks and dependency modeling ability of CRFs, together with a newly designed potential function.

3. PROPOSED MODEL

Our new model presents an extension from existing neural CRFs [43, 44]. Existing neural CRFs in many other sequence labeling tasks are restricted by a fixed set of labels, e.g., *PERSON, LOCATION, ORGANIZATION, MISC* in the name entity recognition (NER) task, and thus can not be applied for open-ontology slot filling. To overcome this shortcoming, we propose a novel framework called elastic conditional random field (eCRF), which consists of three parts. (1) A slot description encoder is employed to encode the slot descriptions into semantic embeddings, then (2) a BiLSTM is used to extract

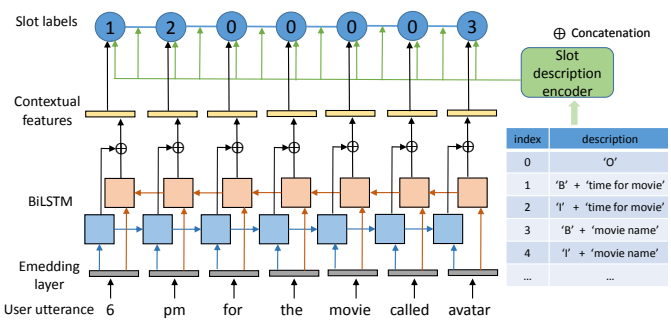

Fig. 2. The architecture of the elastic CRF (eCRF) model.

contextual neural features, and finally (3) the outputs of both the slot description encoder and the BiLSTM are combined to define a novel potential function in the CRF. The main framework of eCRF is illustrated in Figure 2 and each part is detailed in the following subsections.

3.1. Slot Description Encoder

Let $X = (x_1, x_2, ..., x_n)$ denote the input user utterance and $D^i = (d^i_1, d^i_2, ...)$ denote the description of slot s^i. In our experiment, slot descriptions are simple complementary phrases, e.g., 'number of people' for the slot `#people`, 'theatre name' for the slot `theatre_name`, but other richer expression can be used. The goal of our task is to find all possible text spans in X as values for each s^i. We adapted the IOB tagging scheme as in [3]. Traditionally, the IOB tags are made up three type, 'B', 'I', and 'O', which indicate the beginning position of a value span, the intermediate and ending positions of the value span and the rest position belonging to no values. To be specific, if a word is predicted to have the 'B' tag or multiple words are predicted to have '$B, I, .., I$' tags, the word span is the value of a slot. Instead of using a combination of the slot name and one of the IOB tags as in Figure 1, we used the combination of the slot description and one of the IOB tags in order to leverage the semantic meanings of slots. As shown in Figure 2, the slot description encoder takes all slot descriptions as input, and outputs are distributed representations for all possible combinations of the IOB tags and the slot descriptions, such as 'O', '$B + D^1$', '$I + D^1$', '$B + D^2$', '$I + D^2$', ... The set of these new combined slot labels is denoted as $\mathcal{S}$. We use indexes of these labels to suggest the corresponding positions within the utterance. For example, in Figure 2, '6', 'pm' and 'avatar' are predicted as the positions of '$B+$ time for movie', '$I+$ time for movie' and '$B+$ movie name', which means that '6 pm' is the value of slot `movie_time` and 'avatar' is the value of slot `movie_name`. A function $e(\cdot) \in \mathbb{R}^d$ is used to denote the output vector from the slot description encoder as

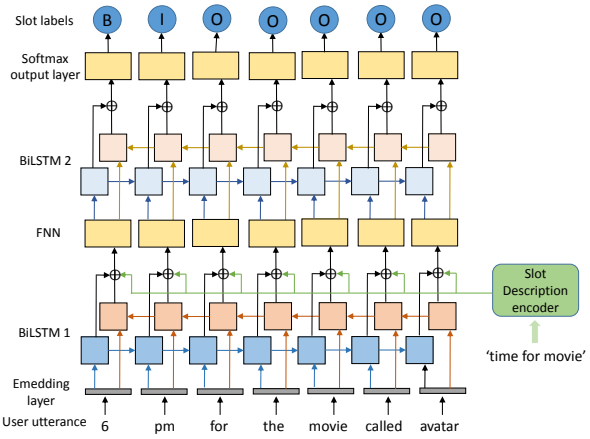

Fig. 3. The architecture of the concept Tagging (CT) model. [12]

follows:

$$e(B + D^i) = FC\left(f(D^i) \oplus emb(B)\right) \quad (1)$$

$$e(I + D^i) = FC\left(f(D^i) \oplus emb(I)\right) \quad (2)$$

$$e(O) = FC\left(\overrightarrow{\mathbf{0}} \oplus emb(O)\right) \quad (3)$$

where $FC(\cdot)$ denotes a one-hidden-layer fully connected network and $f(\cdot)$ denotes an encoder that maps the descriptions into semantic embeddings. In this paper, we use a simple averaging function of all word embeddings in D^i as in [12]. $emb(\cdot)$ is an embedding lookup function for the IOB tags and $\oplus$ denotes the concatenation operation. Note that for $e(O)$, we use a zero vector $\overrightarrow{\mathbf{0}}$ with the same size as the output vector of $f(\cdot)$ since the 'O' tag should be independent of any D^i. A difference between our slot description encoder and that in [12] is that we leverage the embeddings of the IOB tags so that the dependencies between tags in different slot labels are modeled.

3.2. BiLSTM Feature Extractor

Bidirectional long short-term memory (BiLSTM) has been widely utilized in sequence models to capture the contextual semantic feature of input sentences [43, 20]. In eCRF, we also exploit BiLSTMs to extract the contextual neural features. Through concatenating the hidden states from both forward and backward passes, we acquire the distributed representations of contextual features $H = (h_1, h_2, ...h_n)$, in which each $h_i \in \mathbb{R}^d$.

3.3. Elastic CRF (eCRF) Labeler

Let $Y = (y_1, y_2, ...y_n)$ denote the output sequence of slot labels, where $y_i \in \mathcal{S}$. Then the potential function of our elastic neural CRF is defined as follows:

$$\Psi(Y, W) = \sum_{i=1}^{n} e(y_i)^T h_i + \sum_{i=1}^{n-1} e(y_i)^T W e(y_{i+1}) \quad (4)$$

where $W \in \mathbb{R}^{d \times d}$ is a learnable matrix. The potential function consists of two items. The first term, called the node potential, calculates semantic similarity of the slot descriptions and the extracted contextual features. The second term, called the edge potential, captures interactions between the slot labels through a bilinear calculation. Then, the likelihood of eCRF is defined as follows:

$$p(Y|X, D) = \frac{\exp(\Psi(Y, X))}{\sum_{Y'} \exp(\Psi(Y', X))} \quad (5)$$

The eCRF is trained by conditional maximum likelihood (CML), and we used Viterbi decoding for inferences as follows:

$$\hat{Y} = \underset{y'_1,...,y'_n \in \mathcal{S}}{\operatorname{argmax}} \; p(y'_1, ..., y'_n|X, D) \quad (6)$$

In our experiment, we employed the pre-train trick [45] to speed up model learning. Namely, we first masked the edge potential term and trained only with the node potential term for a certain number of training steps, and then added the edge potentials in training. More details can be found in Section 5.2.

4. DATASET AND TASKS

In the experiments, we used the recent Google simulated dataset (accessed from https://github.com/google-research-datasets/simulated-dialogue on 1 June 2018) as our main dataset. It is collected by the machines talking to machines (M2M) self-play schema [13]. Two domains, restaurant and movie, were chosen. There are two common slots, i.e., `time` and `date`, in both domains, and an around 40% out-of-vocabulary (OOV) rate in the test sets. However, since this dataset was not originally built for the open-ontology slot filling, the number of unseen values in the testing set is very limited. In order to properly use this dataset for the study, we designed two different tasks, the in-domain task and the cross-domain task, and accordingly re-split the whole dataset into new training and testing sets.

In the in-domain task, we aimed to evaluate various models for handling unknown values given all known slots. For each domain, we re-split the whole dataset by fixing the ratio between the number of types of values in training and testing. Suppose the sets of all values occurred in the training set and testing set are V_{train} and V_{test}, respectively; we defined the value ratio between training and testing as $|V_{train}| : |V_{test} - V_{train}|$. Three value ratios were chosen for model evaluations, that is, 75:25, 50:50 and 25:75.

For the cross-domain task, we aimed to evaluate various models for handling unknown slots. Similar to the zero-shot multi-domain learning [12], we trained the model on one domain and evaluated it on the other domain. The common slots of the two domains are treated as known slots while the other slots were treated as unknown slots.

After determining the training and testing sets, a validation set is randomly extracted from the training set, satisfying two conditions: (1) the ratio between the total number of utterances in the new training set and validation set is 4:1, and (2) around 50% of the validation set contains unseen slots or values with respect to the new training set. In this way, a reasonable validation set is constructed so that model training can be monitored for stopping for open-ontology prediction.

5. EXPERIMENTS

5.1. Baselines

In this paper, we compare our eCRF model with the concept tagging model proposed in [12] and a simple BiLSTM-based tagging model.

As shown in Figure 3, the Concept tagging (CT) model employs a slot description encoder that takes the slot descriptions as input without the IOB tags. A one-layer BiLSTM is used to extract the contextual features of user utterances. The contextual features and the description encoder outputs are concatenated and sent to a feedforward neural network (FNN). This is followed by another one-layer BiLSTM. Finally, a softmax layer is used to calculate the distribution over slot labels. Since the slot descriptions are already used as conditional inputs, the output slot label set only consists of three labels, i.e., 'I', 'B', 'O'. In both training and testing, the descriptions of each slot are iteratively fed into the model and evaluated separately.

The BiLSTM tagging (BT) model is a simplified version of the CT model, created by removing the second BiLSTM layer. As shown in the following experimental results, this second BiLSTM layer plays an important role in transforming the contextual features and slot label features, which largely improves the performance.

5.2. Experimental Setup

In our experiment, the vocabulary size is 1264. We use the open tool (accessed from https://github.com/stanfordnlp/GloVe on 25 October 2015) to train the GloVe embeddings on the whole dataset. The dimension of all word embeddings and the IOB tags are set as 50. The concatenated hidden size of all BiLSTMs are set as 100. The FNNs in the CT and BT models consist of one hidden layer with 100 units. For the pre-training of eCRFs, the edge potential is added in training after 2000 steps. All models are trained with the Adam [46] optimization method with a learning rate of 0.001. Early-stopping is employed on the validation set to prevent over-fitting. For both the CT and BT models, we leveraged oversampling, which sets the ratio of positive and negative samples as 1:1 and trains the model with a minibatch size of 10. For eCRFs, we set the minibatch size as 1. All the codes were implemented with Tensorflow[47].

Domain	Value-Ratio Train: Test	Average Accuracy for Known Values			Average Accuracy for Unknown Values			Average Accuracy for Total Values		
		BT	CT	eCRF	BT	CT	eCRF	BT	CT	eCRF
sim-R	75:25	0.959±0.020	**0.993±0.005**	0.982±0.007	0.555±0.122	0.753±0.108	**0.791±0.047**	0.765±0.069	0.862±0.060	**0.875±0.026**
	50:50	0.968±0.017	**0.994±0.002**	0.984±0.011	0.361±0.083	0.474±0.066	**0.618±0.058**	0.639±0.048	0.677±0.042	**0.754±0.035**
	25:75	0.967±0.041	**0.999±0.001**	0.985±0.009	0.365±0.034	0.441±0.035	**0.516±0.036**	0.554±0.016	0.575±0.030	**0.624±0.027**
sim-M	75:25	0.951±0.034	0.982±0.005	**0.984±0.003**	0.843±0.009	0.876±0.066	**0.905±0.011**	0.914±0.018	0.930±0.037	**0.953±0.005**
	50:50	0.941±0.028	**0.982±0.009**	0.975±0.017	0.655±0.024	0.723±0.076	**0.841±0.024**	0.803±0.014	0.840±0.040	**0.910±0.017**
	25:75	0.948±0.024	**0.991±0.003**	0.988±0.005	0.519±0.034	0.611±0.030	**0.682±0.035**	0.662±0.027	0.718±0.021	**0.784±0.023**

Table 1. Results for the in-domain tasks: average exact matching accuracies for known values, unknown values and total values for three models. Models are BiLISM tagging (BT) model, concept tagging (CT) model [12] and elastic CRF (eCRF). Sim-R and sim-M are the domains of restaurant and movie respectively. For each domain, three ratios between the number of types of values in training and testing are chosen to re-split the whole dataset to train models. Bold numbers mean the best results among three compared models.

Train Domain	Test Domain	Average Accuracy for Known Slots			Average Accuracy for Unknown Slots			Average Accuracy for Total Slots		
		BT	CT	eCRF	BT	CT	eCRF	BT	CT	eCRF
sim-M	sim-R	0.980±0.025	0.974±0.009	**0.988±0.004**	0.136±0.045	0.121±0.077	**0.243±0.009**	0.502±0.036	0.491±0.044	**0.566±0.007**
sim-R	sim-M	0.814±0.064	0.915±0.013	**0.926±0.024**	0.165±0.040	0.246±0.017	**0.377±0.031**	0.508±0.035	0.599±0.006	**0.667±0.020**

Table 2. Results for the cross-domain tasks: average exact matching accuracies for values from known slots, unknown slots and total slots on test domain for three models. Bold numbers mean the best results among three compared models.

5.3. In-Domain Task Results

As described in Section 4, for the in-domain tasks, we re-organized the whole dataset into three different new datasets with increasing prediction difficulties, by setting the value ratios between training and testing as 75:25, 50:50 and 25:75. Table 1 shows the average exact-matching accuracies for known values, unknown values, and total values on the testing set for each model.

The results demonstrate that eCRFs clearly outperform the BT models in all conditions. Though slightly worse than the CT models on known values, eCRFs achieve much better results than the CT models in terms of accuracies for unknown values. And the superiority becomes larger as the value-ratio in testing set becomes higher. Therefore, in terms of accuracies for total values, eCRFs achieve the best overall performances.

5.4. Cross-Domain Task Results

For the cross-domain tasks, we train models on one domain and test on the other. The common slots such as `time`, `date` are treated as known slots while the rest as unknown slots, such as `theatre_name`, `restaurant_name`. The evaluation metrics are the average exact-matching accuracies for values from known slots, unknown slots and total slots on the target domain. As shown in Table 4, eCRFs outperform other models in all conditions. In the cross-domain tasks, although there are some overlapping between the known slots on the two domains, the user utterances are different in expressing those slots and values. These results demonstrate that our eCRFs have greater generalization ability.

Figures 4–6 show the prediction results for the same utterance on the movie domain with the eCRF and CT models. Figure 4 illustrates the predicted scores with only node potentials for eCRFs, while Figure 5 gives the predicted scores with both node and edge potentials. It can be seen that the boundaries of slot labels for some slots are mistakenly placed in Figure 4, e.g., the value "*lincoln square cinemas*" for the unknown slot `theatre_name` is falsely predicted as two values "*lincoln*" and "*square cinemas*". When taking both node and edge potentials into account, correct predictions are obtained for all the three slots, as shown in Figure 5. The output probabilities of slot labels for the CT model are shown in Figure 6. Although the CT model gives the right prediction for the known slot `date` and unknown slot `#tickets`, it mistakenly predicts the value for the unknown slot `theatre_name` as "*lincoln square*", as it fails to learn the semantic relations between slot labels.

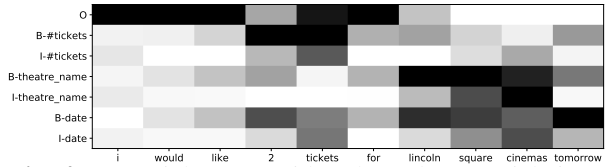

Fig. 4. Potential scores with only node potentials in eCRFs for the cross-domain task. The darker the color, the higher the potential score.

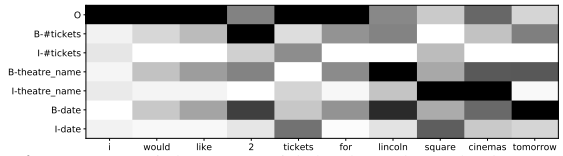

Fig. 5. Potential scores with both node and edge potentials in eCRFs for the cross-domain task.

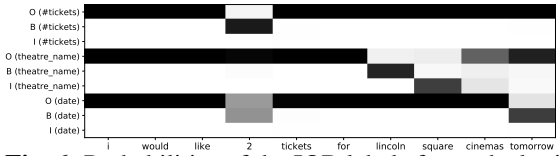

Fig. 6. Probabilities of the IOB labels for each slot in the CT model.

6. CONCLUSIONS

In this paper, we propose a novel model, the elastic conditional random field (eCRF), for open-ontology slot-filling task. The natural language descriptions of slots and (user) utterances are encoded into the same semantic embedding space to implement the node and edge potentials. We recompose the Google simulated dataset and demonstrate that eCRFs achieve better performances in both in-domain tasks and cross-domain tasks than existing models.

There are interesting future works to further enhance the parsing ability and adaptation capacity of eCRFs: (1) encoding the descriptions of more semantic labels including the intent labels, domain labels and action labels for better generalization and (2) upgrading the CRF architecture with a slot label language model that can capture long-range dependencies between labels.

7. REFERENCES

[1] Ye Yi Wang, Li Deng, and A Acero, "Spoken language understanding," *Signal Processing Magazine IEEE*, vol. 22, no. 5, pp. 16–31, 2005.

[2] Grgoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, and Dong Yu, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 23, no. 3, pp. 530–539, 2015.

[3] L. A. Ramshaw and M. P. Marcus, *Text Chunking Using Transformation-Based Learning*, Springer Netherlands, 1999.

[4] John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[5] Bing Liu and Ian Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Interspeech*, 2016.

[6] Erik F Tjong Kim Sang and Fien De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *HLT-NAACL*, 2003.

[7] Bing Liu and Ian 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] Abhinav Rastogi, Dilek Z. Hakkani-Tür, and Larry P. Heck, "Scalable multi-domain dialogue state tracking," *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 561–568, 2017.

[9] Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun Nung Chen, Jianfeng Gao, Li Deng, and Ye Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *InterSpeech*, 2016.

[10] Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Pei hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young, "Multi-domain dialog state tracking using recurrent neural networks," in *ACL*, 2015.

[11] Aaron Jaech, Larry P. Heck, and Mari Ostendorf, "Domain adaptation of recurrent neural networks for natural language understanding," in *INTERSPEECH*, 2016.

[12] Ankur Bapna, Gökhan Tür, Dilek Z. Hakkani-Tür, and Larry P. Heck, "Towards zero-shot frame semantic parsing for domain scaling," in *INTERSPEECH*, 2017.

[13] Pararth Shah, Dilek Z. Hakkani-Tür, Gökhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry P. Heck, "Building a conversational agent overnight with dialogue self-play," *CoRR*, 2018.

[14] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio, "Zero-data learning of new tasks," in *AAAI*, 2014.

[15] Lina Rojas-Barahona, Bo-Hsiang Tseng, Yinpei Dai, Clare Mansfield, Osman Ramadan, Stefan Ultes, Michael Crawford, and Milica Gasic, "Deep learning for language understanding of mental health concepts derived from cognitive behavioural therapy," *arXiv preprint arXiv:1809.00640*, 2018.

[16] Yun Nung Chen, Dilek Hakkani-Tür, and Xiaodong He, "Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models," in *ICASSP*, 2016.

[17] Tiancheng Zhao and Maxine Eskénazi, "Zero-shot dialog generation with cross-domain latent actions," in *SIGDIAL*, 2018.

[18] Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek 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, 2019, pp. 5484–5490, Association for Computational Linguistics.

[19] Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba, "Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021, pp. 5640–5648, Association for Computational Linguistics.

[20] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang, "Improving sentiment analysis via sentence type classification using bilstm-crf and cnn," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[21] Puyang Xu and Ruhi Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *ASRU*, 2014, pp. 78–83.

[22] Yinpei Dai, Wanwei He, Bowen Li, Yuchuan Wu, Zheng Cao, Zhongqi An, Jian Sun, and Yongbin Li, "Cgodial: A large-scale benchmark for chinese goal-oriented dialog evaluation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 4097–4111.

[23] Shuzheng Si, Wentao Ma, Yuchuan Wu, Yinpei Dai, Haoyu Gao, Ting-En Lin, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li, "Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue in multiple domains," *arXiv preprint arXiv:2305.13040*, 2023.

[24] Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu, "Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 879–885.

[25] Yichi Zhang, Yinpei Dai, Zhijian Ou, Huixin Wang, and Junlan Feng, "Improved learning of word embeddings with word definitions and semantic injection.," in *INTERSPEECH*, 2020, pp. 4253–4257.

[26] Yinpei Dai, Zhijian Ou, Dawei Ren, and Pengfei Yu, "Tracking of enriched dialog states for flexible conversational information access," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6139–6143.

[27] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu, "Leveraging sentence-level information with encoder lstm for semantic slot filling," in *EMNLP*, 2016.

[28] Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze, "Bi-directional recurrent neural network with ranking loss for spoken language understanding," in *ICASSP*, 2016.

[29] Yinpei Dai, Huihua Yu, Yixuan Jiang, Chengguang Tang, Yongbin Li, and Jian Sun, "A survey on dialog management: Recent advances and challenges," *arXiv preprint arXiv:2005.02233*, 2020.

[30] Yun Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng, "End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding," in *InterSpeech*, 2016.

[31] Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung, "Coach: A coarse-to-fine approach for cross-domain slot filling," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 19–25, Association for Computational Linguistics.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[33] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li, "A unified mrc framework for named entity recognition," *arXiv preprint arXiv:1910.11476*, 2019.

[34] Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Hakkani-Tur, "From machine reading comprehension to dialogue state tracking: Bridging the gap," *arXiv preprint arXiv:2004.05827*, 2020.

[35] Mengshi Yu, Jian Liu, Yufeng Chen, Jinan Xu, and Yujie Zhang, "Cross-domain slot filling as machine reading comprehension," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou, Ed. 8 2021, pp. 3992–3998, International Joint Conferences on Artificial Intelligence Organization, Main Track.

[36] Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne, "Transferable dialogue systems and user simulators," *arXiv preprint arXiv:2107.11904*, 2021.

[37] Yinpei Dai, Hangyu Li, Chengguang Tang, Yongbin Li, Jian Sun, and Xiaodan Zhu, "Learning low-resource end-to-end goal-oriented dialog for fast and reliable system deployment," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 609–618.

[38] Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei

Huang, Luo Si, et al., "Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection," in *Proceedings of the AAAI conference on artificial intelligence*, 2022, vol. 36, pp. 10749–10757.

[39] Wanwei He, Yinpei Dai, Binyuan Hui, Min Yang, Zheng Cao, Jianbo Dong, Fei Huang, Luo Si, and Yongbin Li, "Space-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 553–569.

[40] Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li, "Unified dialog model pre-training for task-oriented dialog understanding and generation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 187–200.

[41] Weizhi Wang, Zhirui Zhang, Junliang Guo, Yinpei Dai, Boxing Chen, and Weihua Luo, "Task-oriented dialogue system as natural language generation," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 2698–2703.

[42] Haitao Mi, Qiyu Ren, Yinpei Dai, Yifan He, Jian Sun, Yongbin Li, Jing Zheng, and Peng Xu, "Towards generalized models for beyond domain api task-oriented dialogue," in *AAAI-21 DSTC9 Workshop*, 2021.

[43] Xuezhe Ma and Eduard Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *ACL*, 2016.

[44] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer, "Neural architectures for named entity recognition," in *NAACL-HLT*, 2016.

[45] David Belanger and Andrew McCallum, "Structured prediction energy networks," in *ICML*, 2016.

[46] Jimmy. Kingma, Diederik & Ba, "Adam: A method for stochastic optimization.," *International Conference on Learning Representations*, 2014.

[47] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard, "Tensorflow: A system for large-scale machine learning," in *OSDI*, 2016.