SlotRefine: A Fast Non-Autoregressive Model for Joint Intent Detection and Slot Filling

Di Wu[§]   **Liang Ding**[†]   **Fan Lu**[♭]   **Jian Xie**[‡]

[§]Peking University   [♭]University of Washington   [‡]Wuhan University

`inbath@163.com`   `lufan0929@gmail.com`   `xiejian1990@gmail.com`

[†]UBTECH Sydney AI Centre, School of Computer Science

Faculty of Engineering, The University of Sydney

`ldin3097@uni.sydney.edu.au`

Abstract

Slot filling and intent detection are two main tasks in spoken language understanding (SLU) system. In this paper, we propose a novel non-autoregressive model named SlotRefine for joint intent detection and slot filling. Besides, we design a novel two-pass iteration mechanism to handle the uncoordinated slots problem caused by conditional independence of non-autoregressive model. Experiments demonstrate that our model significantly outperforms previous models in slot filling task, while considerably speeding up the decoding (up to ×10.77). In-depth analyses show that 1) pretraining schemes could further enhance our model; 2) two-pass mechanism indeed remedy the uncoordinated slots.

1 Introduction

Slot filling (SF) and intent detection (ID) play important roles in spoken language understanding, especially for task-oriented dialogue system. For example, for an utterance like "*Buy an air ticket from Beijing to Seattle*", intent detection works on sentence-level to indicate the task is about purchasing an air ticket, while the slot filling focus on words-level to figure out the departure and destination of that ticket are "*Beijing*" and "*Seattle*".

In early studies, ID and SF were often modeled separately, where ID was modeled as a classification task, while SF was regarded as a sequence labeling task. Due to the correlation between these two tasks, training them jointly could enhance each other. Zhang and Wang (2016) propose a joint model using bidirectional gated recurrent unit to learn the representation at each time step. Meanwhile, a max-pooling layer is employed to capture the global features of a sentence for intent classification. Liu and Lane (2016) cast the slot filling task as a tag generation problem and introduce a recurrent neural network based encoder-decoder framework with attention mechanism to model it, meanwhile using the encoded vector to predict intent. Goo et al. (2018) and Haihong et al. (2019) dig into the correlation between ID and SF deeper and modeled the relationship between them explicitly. Qin et al. (2019) propagate the token-level intent results to the SF task, achieving significant performance improvement.

Briefly summarized, most of the previous works heavily rely on autoregressive approaches, *e.g.,* RNN based model or seq2seq architecture, to capture the grammar structure in an utterance. And conditional random field (CRF) is a popular auxiliary module for SF task as it considers the correlations between tags. Thus, several state-of-the-art works combine the autoregressive model and CRF to achieve the competitive performance, which therefore are set as our baseline methods.

However, for SF task, we argue that identifying token dependencies among slot chunk is enough, and it is unnecessary to model the entire sequence dependency in autoregressive fashion, which leads to redundant computation and inevitable high latency.

In this study, we cast these two tasks jointly as a non-autoregressive tag generation problem to get rid of unnecessary temporal dependencies. Particularly, a Transformer (Vaswani et al., 2017) based architecture is adopted here to learn the representations of an utterance in both sentence and word level simultaneously (Sec.§2.1). The slots and intent labels are predicted independently and simultaneously, achieving better decoding efficiency. We further introduce a two-pass refine mechanism (in Sec.§2.2) to model boundary prediction of each slots explicitly, which also handle the uncoordinated slots problem (*e.g., I-song* follows *B-singer*) caused by conditional independence attribute.

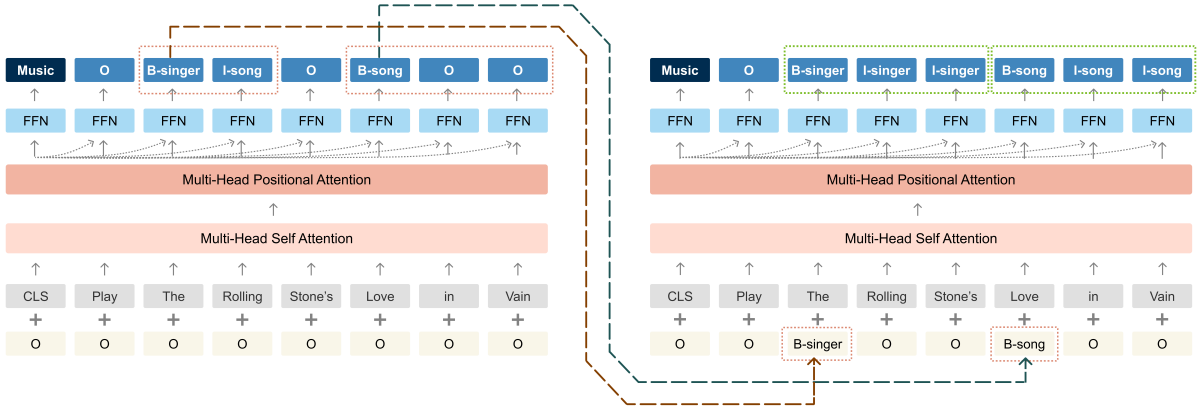Experiments on two commonly-cited datasets

Figure 1: Illustration of SlotRefine, where the left and right part indicate the first and second iteration process respectively. In the first pass, wrong slot tagging results are predicted, as shown in the pink dotted box in the figure, and the "*B-tags*" (beginning tag of a slot) are feed as additional information with utterance for second iteration. The slot results in the green dotted box are refined results by second pass. Note that the initial tag embedding "O" added to each inputting position is designed for the two-pass mechanism(Sec.§2.2).
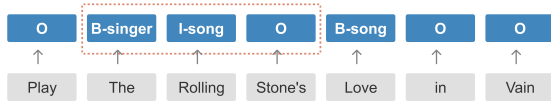

Figure 2: A example of uncoordinated slot tagging.

show that our approach is significantly and consistently superior to the existing models both in SF performance and efficiency (Sec.§3). Our contributions are as follows:

- We propose an fast non-autoregressive approach to model ID and SF tasks jointly, named SlotRefine[1], achieving the state-of-the-art on ATIS dataset.

- We design a two-pass refine mechanism to handle uncoordinated slots problem. Our analyses confirm it is a better alternative than CRF in this task.

- Our model infers nearly $\times 11$ faster than existing models ($\times 13$ for long sentences), indicating that our model has great potential for the industry and academia.

2 Proposed Approaches

In this section, we first describe how we model slot filling and intent detection task jointly by an non-autoregressive model. And then we describe the details of the two-pass refine mechanism. The brief

scheme of our model is shown in Figure 1, details can be found in the corresponding caption. Note that we follow the common practice (Ramshaw and Marcus, 1995; Zhang and Wang, 2016; Haihong et al., 2019) to use "Inside–outside–beginning (IOB)" tagging format.

2.1 Non-Autoregressively Joint Model

We extend the original multi-head Transformer encoder in Vaswani et al. (2017) to construct the model architecture of SlotRefine. Please refer to Vaswani et al. (2017) for the details of Transformer. The main difference against the original Transformer is that we model the sequential information with relative position representations (Shaw et al., 2018), instead of using absolute position encoding.

For a given utterance, a special token CLS is inserted to the first inputting position akin to the operation in BERT (Devlin et al., 2019). Difference from that in BERT is the corresponding output vector is used for next sentence classification, we use it to predict the label of intent in SlotRefine. We denote the input sequence as $x = (x_{cls}, x_1, ..., x_l)$, where l is the utterance length. Each word x_i will be embedded into a h-dimension vector to perform the multi-head self-attention computation. Then, the output of each model stack can be formulated as $H = (h_{cls}, h_1, ..., h_l)$.

To jointly model the representations of ID and SF tasks, we directly concat [2] the representations of

---

[2]We follow (Goo et al., 2018) to fuse two representations with gating mechanism, but preliminary experiments show that simply concatenation performs best for our model structure.

Model	ATIS Dataset			Snips Dataset		
	Slot	**Intent**	**Sent**	**Slot**	**Intent**	**Sent**
Joint Seq (Hakkani-Tür et al., 2016)	94.30	92.60	80.70	87.30	96.90	73.20
Atten.-Based (Liu and Lane, 2016)	94.20	91.10	78.90	87.80	96.70	74.10
Sloted-Gated (Goo et al., 2018)	95.42	95.41	83.73	89.27	96.86	76.43
SF-ID (w/o CRF) (Haihong et al., 2019)	95.50	96.58	86.00	90.46	97.00	78.37
SF-ID (w/ CRF) (Haihong et al., 2019)	95.80	97.09	86.90	92.23	97.29	80.43
Stack-Propagation (Qin et al., 2019)	95.90	96.90	86.50	94.20	98.00	86.90
Our Joint Model (in Sec.§2.1)	95.33	96.84	85.78	93.13	97.21	82.83
Our Joint Model +CRF	95.71	96.54	85.71	93.22	96.79	82.51
SlotRefine	**96.22**$^{\uparrow}$	**97.11**$^{\uparrow}$	**86.96**$^{\uparrow}$	**93.72**	97.44	84.38

Table 1: Performance comparison on ATIS and Snips datasets. "↑"indicates significant difference ($p < 0.05$) with previous works. Model name written in bold refer to ours.

h_{cls} and h_i before feed-forward computation, and then feed them into the softmax classifier. Specifically, the intent detection and slot filling results are predicted as follows, respectively:

$$y^i = \text{softmax}\left(W^i \cdot h_{cls} + b^i\right)$$
$$y_i^s = \text{softmax}\left(W^s \cdot [h_{cls}, h_i] + b^s\right) \quad (1)$$

where y^i and y_i^s denote intent label of the utterance and slot label for each token i, respectively. $[h_{cls}, h_i]$ is the concated vector. W and b are corresponding trainable parameters.

The objective of our joint model can be formulated as:

$$p\left(y^i, y^s|x\right) = p\left(y^i|x\right) \cdot \prod_t^l p\left(y_t^s|x, y^i\right) \quad (2)$$

The learning objective is to maximize the conditional probability $p\left(y^i, y^s|x\right)$, which is optimized via minimizing its cross-entropy loss. Unlike autoregressive methods, the likelihood of each slot in our approach can be optimized in parallel.

2.2 Two-pass Refine Mechanism

Due to the conditional independence between slot labels, it is difficult for our proposed non-autoregressive model to capture the sequential dependency information among each slot chunk, thus leading to some uncoordinated slot labels. We name this problem as **uncoordinated slots problem**. Take the false tagging in Figure 2 for example, slot label "*I-song*" uncoordinately follows "*B-singer*", which does not satisfy the Inside-Outside-Beginning tagging format.

To address this problem, we introduce a two-pass refine mechanism. As depicted in the Figure 1, in addition to each token embedding in the utterance, we also element-wisely add the slot tag embedding

into the model. In the first pass, the initial slot tags are all setting to "O", while in the second pass, the "*B-tags*" predicted in the first pass is used as the corresponding slot tag input. These two iterations share the model and optimization goal, thus brings no extra parameters.

Intuitively, in doing so, the model generates a draft in the first pass and tries to find the beginning of each slot chunk. In the second pass, by propagating the utterance again with the predicted "*B-tags*", the model is forced to learn how many identical "*I-tags*" follow them. Through this process, the slot labels predicted becomes more consistent, and the boundaries are more accurately identified. From a more general perspective, we can view this two-pass process as a trade-off between autoregression and non-autoregression, where the complete markov chain process can be simplified as follow:

$$p\left(y^i, y^s|x\right) = p\left(y^i|x\right) \cdot p\left(\tilde{y}^s|y^i, x\right)$$
$$\cdot p\left(y^s|\tilde{y}^s, y^i, x\right) \quad (3)$$

where $\tilde{y}^s$ is the tagging results from the first pass.

Two-pass refine mechanism is similar to the multi-round iterative mechanism in non-autoregressive machine translation (Lee et al., 2018; Gu et al., 2018; Ding et al., 2020; Kasai et al., 2020), such as Mask-predict (Ghazvininejad et al., 2019). However, we argue that our method is more suitable in this task. The label dependency of the tagging task (e.g., slot filling) is simple, where we only need to ensure the tagging labels of a slot are consistent from the beginning to the end. Therefore, two iterations to force the model to focus on the slot boundaries is enough in our task, intuitively. Mask-Predict can alleviate the problem caused by conditional independence too. However, it's designed for a more complex goal, and it usually

introduce more iterations (e.g., 10 iters) to achieve competitive performance, which largely reduces the inference speed.

3 Experiment

Datasets We choose two widely-used datasets: ATIS (Airline Travel Information Systems,Tur et al. (2010)) and Snips (collected by Snips personal voice assistant,Coucke et al. (2018)). Compared with ATIS, the Snips dataset is more complex due to its large vocabulary size, cross-domain intents and more out-of-vocabulary words.

Metrics Three evaluation metrics are used in our experiments. F1-score and accuracy are applied for slot filling and intent detection task, respectively. Besides, we use sentence accuracy to indicate proportion of utterance in the corpus whose slots and intent are both correctly-predicted.

Setup All embeddings are initialized with xavier method (Glorot and Bengio, 2010). The batch size is set to 32 and learning rate is 0.001. we set number of Transformer layers, attention heads and hidden sizes to {2,8,64} and {4,16,96} for ATIS and Snips datasets. In addition, we report the results of previous studies (Hakkani-Tür et al., 2016; Liu and Lane, 2016; Goo et al., 2018; Haihong et al., 2019; Qin et al., 2019) and conduct speed evaluation based on their open-source codes.

Main Results Table 1 summarizes the model performance on ATIS and snips corpus. It can be seen that SlotRefine consistently outperforms other baselines in all three metrics. Compared with our basic non-autoregressive joint model in Section§ 2.1, SlotRefine achieve +1.18 and +1.55 sentence-level accuracy improvements for ATIS and Snips, respectively. It is worthy noting that our SlotRefine significantly improves the slot filling task (F1-score↑). we attribute the improvement to that our two-pass mechanism successfully makes the model learn better slot boundaries.

Speedup As each slot tagging result can be calculated in parallel with our approach, inference latency can be significantly reduced. As shown in Table 2, on ATIS test set, our non-autoregressive model could achieve ×8.80 speedup compared with the existing state-of-the-art model (Haihong et al., 2019). And after introducing two-pass mechanism (SlotRefine), our model still achieves competitive inference speedup (×4.31). Our decoding

Model	Latency	Speedup
Sloted-Gated	11.31ms	1.41×
SF-ID (with CRF)	13.03ms	1.22×
Stack-Propagation	15.94ms	1.00×
Our Joint Model	1.48ms	10.77×
Our Joint Model +CRF	8.32ms	1.92×
SlotRefine	3.02ms	4.31×

Table 2: "Latency" is the average time to decode an utterance without minibatching. "Speedup" is compared against existing SOTA model (Haihong et al., 2019).
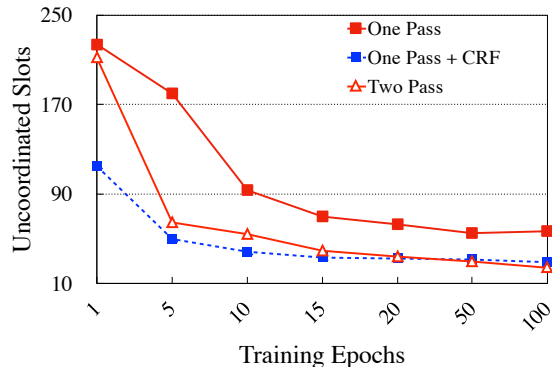

Figure 3: The number of uncoordinated slots of our joint model (One Pass), joint model with CRF (One Pass+CRF) and SlotRefine (Two Pass) during training.

is conducted with a single Tesla P40 GPU. It is worth noting that for long sentences (Length≥12), the speedup achieves ×13 (not reported in table).

Two-Pass Mechanism v.s. CRF In SF task, CRF is usually used to learn the dependence of slot labels. Two most important dependence rules CRF learned can be summarized as *tag O can only be followed by O or B* and *tag B-* can only be followed by same-type label I-* or O*, which can be perfectly addressed with our proposed two-pass mechanism. Experiments about +CRF can be found in Table 1&2 ("Our Joint Model +CRF"), we can see that two-pass mechanism equipped SlotRefine outperforms +CRF by averagely +0.89, meanwhile preserving ×2.8 speedup, demonstrating that two-pass mechanism can be a better substitute for CRF in this task for better performance and efficiency.

Remedy Uncoordinated Slots in Training We visualize the number decrease of uncoordinated slots of the training process on ATIS dataset. As depicted in Figure 3, uncoordinated errors of both "One-Pass" and "Two-Pass" models decrease with training goes. Notably, the uncoordinated slots number of Two-Pass model drops significantly

Model	ATIS Dataset			Snips Dataset		
	Slot	**Intent**	**Sent**	**Slot**	**Intent**	**Sent**
Joint Model	95.33	96.84	85.78	93.31	97.21	82.83
Joint Model with CRF	95.71	96.54	85.71	93.22	96.79	82.51
SlotRefine	96.22	97.11	86.96	93.72	97.44	84.38
SlotRefine with GloVe	**96.24**	97.35	87.57	96.33	98.36	91.06
SlotRefine with BERT	96.16	**97.74**	**88.64**	**97.05**	**99.04**	**92.96**
previous work with pretraining						
BERT-Joint (Chen et al., 2019)	96.10	97.50	88.20	97.00	98.60	92.80
Stack-Propagation with BERT (Qin et al., 2019)	96.10	97.50	88.60	97.00	99.00	92.90

Table 3: Performance comparison between SlotRefine with GloVe initialzation and Bert based model on ATIS and Snips datasets.

faster than the One-Pass model, achieving better convergence than +CRF after 50 epochs. This indicates that our proposed two-pass mechanism indeed remedy the uncoordinated slots problem, making the slot filling more accurate.

SlotRefine with Pretraining   Recently, there are also some works based on large scale pretraining model BERT (Chen et al., 2019), where billions of external corpus are used and tremendous of model parameters are introduced. The number of parameters of BERT is many orders of magnitude more than ours, thus it is unfair to compare performance of SlotRefine with them directly. To highlight the effectiveness of SlotRefine, we conduct experiments with two pretraining schemes, GloVe[3] and BERT[4], to compare with them. We find that both GloVe and BERT could further enhance the SlotRefine, and it worth noting that "SlotRefine w/ BERT" outperforms existing pretraining based models. The detailed comparison can be found in Table 3.

For the pre-training scheme of BERT, we follow the setting in Chen et al. (2019) and equip two-pass mechanism in the fine-tune stage, where CLS token is used for intent detection. And for the pre-training scheme of GloVe, we fix and compress the pretrained word vectors into the same dimension of the input hidden size in SlotRefine by a dense network. It is worth noting that through such simple pre-training method, SlotRefine can achieve a results very close to the method implemented by BERT. We guess that the benefits of the pre-training methods on this task mainly come from alleviating the Out-of-Vocabulary (OOV) problem. One

piece of evidence is, for Snips whose test set has a large number of OOV words, benefits through pre-training are very obvious. However, for the ATIS whose test set has few OOV words, only a small sentence accuracy gain, 0.61 and 1.68 for GloVe and Bert respectivly, is obtained after using the pre-training method.

4   Conclusion

In this paper, we first reveal an *uncoordinated slots problem* for a classical language understanding task, i.e., slot filling. To address this problem, we present a novel non-autoregressive joint model for slot filling and intent detection with two-pass refine mechanism (non-autoregressive refiner), which significantly improves the performance while substantially speeding up the decoding. Further analyses show that our proposed non-autoregressive refiner has great potential to replace CRF in at least slot filling task.

In the future, we plan to extend our non-autoregressive refiner to other Natural Language Understanding (NLU) tasks, e.g., named entity recognition (Tjong Kim Sang and De Meulder, 2003), semantic role labeling (He et al., 2018), and Natural Language Generation (NLG) tasks, e.g., machine translation (Vaswani et al., 2017), summarization (Liu and Lapata, 2019).

Acknowledgments

---

[3]https://github.com/stanfordnlp/GloVe
[4]https://github.com/huggingface/transformers

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. In *arXiv*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. In *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Liang Ding, Di Wu, Longyue Wang, Dacheng Tao, and Zhaopeng Tu. 2020. Localness matters: The evolved cross-attention for non-autoregressive translation. In *COLING*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *ICML*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Parallel machine translation with disentangled context transformer. In *ICML*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *EMNLP*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *NAACL*.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*.

1937