PROGRESSIVE DIALOGUE STATE TRACKING FOR MULTI-DOMAIN DIALOGUE SYSTEMS

Jiahao Wang[⋆] *Minqian Liu*[†] *Xiaojun Quan*[⋆‡]

[⋆] School of Computer Science and Engineering, Sun Yat-sen University
[†] School of Computer Science and Engineering, South China University of Technology

ABSTRACT

There are two critical observations in multi-domain dialogue state tracking (DST) ignored in most existing work. First, the number of triples (domain-slot-value) in dialogue states generally increases with the growth of dialogue turns. Second, although dialogue states are accumulating, the difference between two adjacent turns is steadily minor. To model the two observations, we propose to divide the task into two successive procedures: progressive domain-slot tracking and shrunk value prediction. Specifically, domain-slot pairs are first modeled in a multi-level structure that can be predicted progressively based on previous turns. Then, we employ a generative approach to producing dialogue values for the predicted, rather than for all possible, domain-slot pairs. This divide-and-conquer approach not only enables parallelization for predicting domain-slot pairs, but also significantly reduces the number of domain-slot candidates for value prediction. Experimental results on the MultiWOZ datasets confirm that our methodology (PRO-DST) achieves very favorable improvement over existing methods.

Index Terms— Task-Oriented Dialogue Systems, Dialogue State Tracking, Attention

1. INTRODUCTION

Task-oriented dialogue systems aim to facilitate people with such services as taxi booking, food ordering, and hotel reservation through multi-turn natural language conversations. To generate fluent and informative responses, the systems typically use a dialogue state tracker that keeps close track of the dialogue states to manage information about the tasks. The dialogue states are usually organized in triples such as `domain-slot-value`.

Most existing efforts for DST are based on three strategies: candidate classification [1, 2, 3, 4], span prediction [5, 6], and text generation [7, 8, 9]. In the candidate classification strategy, dialogue states are predicted via computing a distribution over all domain-slot-value triples, whereas the number of candidate triples is usually huge. Span prediction directly predicts the start and end positions of dialogue value tokens, yet such approaches may suffer from inconsistent definitions of value between utterances and that of predefined. To address these issues, the generation strategy regards each dialogue value as a string that can be generated by a sequence-to-sequence model.

Recently, pre-trained language models such as BERT [10] have also been applied to DST and achieved impressive performance [5, 6, 11]. However, using these large pre-trained models is computationally costly and might not be practical in some real-life settings. How to build an efficient algorithm for DST remains an open question.

Our work is inspired by two critical observations in DST. The first is `accumulating state triples`, which means the number of triples in dialogue states increases with the growth of dialogue turns. It reflects an inherent challenge in DST that the difficulty will increase as the dialogue gets longer. The second is `adjacent state dependencies`, suggesting that although the states are accumulating, the difference between two adjacent turns is constantly small. Empirical evidence also confirms that users tend not to change their requests sharply between two adjacent turns, as shown in the experiments (Figure 3).

Unfortunately, the observations have been ignored in most existing studies. To fill the void, we propose a model called PRO-DST which separates the multi-domain DST task into two successive procedures: progressive domain-slot tracking and shrunk value prediction. First, domain-slot pairs are modeled in a multi-level structure that facilitates the current turn's prediction to incorporate previous predictions progressively. Second, we employ a generator to produce values for the predicted, rather than for all possible, domain-slot pairs.

The contributions of this paper are summarized as follows: **(1)** We propose to divide DST into two successive stages, *i.e.*, progressive domain-slot tracking and shrunk value prediction, based on our two observations. **(2)** We adopt three levels of embeddings and attentions in our domain-slot tracker to model the domain-slot structure and capture the information on different levels. **(3)** The progressive tracker is able to predict domain-slot pairs in parallel and reduce the number of domain-slot candidates significantly for value prediction, making our model more scalable and efficient.

‡ Corresponding author.

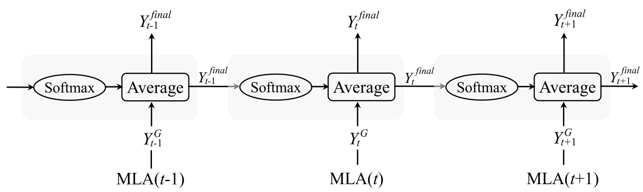

Fig. 1. Overview of our progressive domain-slot tracker module. `MLA` denotes the multi-level attention module, Y_t^G is the output of this module, and Y_t^{final} is the output of our tracker at turn t.

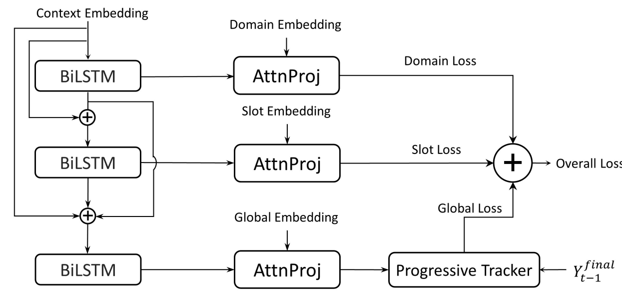

Fig. 2. Illustration of the multi-level attention (MLA) module.

2. METHOD

Our two-stage PRO-DST model is composed of a progressive domain-slot tracking process and a shrunk value generation process. The progressive domain-slot tracker classifies each domain-slot pair from three categories: `pointed`, `dontcare`, and `none`. Then we generate values for those pairs predicted with `pointed`. The pairs predicted with `dontcare` will be assigned with a `dontcare` value directly, while the rest will be labeled as `none`.

We design our domain-slot tracker to contain a multi-level attention (MLA) module and a progressive state tracker. While the MLA module is devised to deal with the dialogue context, the progressive state tracker combines the prediction of previous turns. The architecture of this domain-slot tracker is shown in Figure 1.

2.1. Multi-Level Attention

In multi-domain DST, a slot may belong to more than one domain, domain-slot pairs can be naturally organized in a hierarchical structure. Inspired by the research on hierarchical multi-label classification [12], we design the MLA module based on domain-level, slot-level, and global (the combination of domains and slots)-level classifiers with attention mechanisms as shown in Figure 2.

We denote the embeddings of domains, slots, and domain-slot combinations as $E^D \in \mathbb{R}^{m\times d_e}$, $E^S \in \mathbb{R}^{m\times d_e}$ and $E^G \in \mathbb{R}^{m\times d_e}$, respectively, where m is the number of all candidate domain-slot pairs and d_e is the dimension of word embeddings. Since the number of domains and slots are smaller than m, we duplicate each of their embeddings to size m.

We use $X \in \mathbb{R}^{n\times d_e}$ to denote the word embeddings of a dialogue context, and employ bidirectional long-short term memory (BiLSTM [13]) to derive its encoded representation U, where $U \in \mathbb{R}^{n\times d_h}$ is obtained by summing up the bidirectionally encoded representations. We then calculate a similarity matrix $A^D \in \mathbb{R}^{m\times n}$ for the domain embeddings E^D and the encoded dialogue context U (d_h is set equal to d_e):

$$A^D = E^D U^\mathsf{T}. \tag{1}$$

For the i-th domain, we calculate its attention scores with the encoded context U, and obtain a domain-attended context $U_i^D \in \mathbb{R}^{n\times d_e}$ as:

$$U_i^D = \mathbf{e}(\mathrm{Softmax}(A_i^D)) \odot U, \tag{2}$$

where **e** is the operation to expand A_i^D to the dimension of U, and $\odot$ denotes element-wise multiplication.

Finally, we apply a feed forward network and a softmax layer to the domain-attended context $U^D \in \mathbb{R}^{m\times n\times d_h}$, and obtain a domain-level output distribution $Y^D \in \mathbb{R}^{m\times 3}$:

$$R^D = \mathrm{ReLU}((U^D)W_1^\mathsf{T} + b_1), \tag{3}$$

$$Y_i^D = \mathrm{Softmax}((\sum_{j=1}^{n} R_{ij}^D)W_2^\mathsf{T} + b_2), \tag{4}$$

where $W_1 \in \mathbb{R}^{d_h\times d_h}$, $W_2 \in \mathbb{R}^{3\times d_h}$, $b_1 \in \mathbb{R}^{d_h}$ and $b_2 \in \mathbb{R}^3$ are the parameters to train, and $Y_i^D \in \mathbb{R}^3$ corresponds to the categories of `pointed`, `dontcare`, and `none`.

For simplicity, we use a function $\mathrm{AttnProj(E, C)}$ to denote the equations (1)-(4) above, where E is the domain-level, slot-level or global-level embeddings, and C is the encoded dialogue context. The slot-level output Y^S and the global-level output Y^G are calculated as follows:

$$Y^S = \mathrm{AttnProj}(E^S, \mathrm{BiLSTM}^S(X + U)), \tag{5}$$

$$Y^G = \mathrm{AttnProj}(E^G, \mathrm{BiLSTM}^G(X{+}U{+}V)). \tag{6}$$

Here, $V = \mathrm{BiLSTM}^S(X + U)$ denote the encoded context for slot-level attention and global-level encoding, and the parameters of three BiLSTMs are not shared. Y^D, Y^S, and Y^G are to be used to calculate domain-level, slot-level, and global-level losses, respectively. Moreover, Y^G will also be used by our tracker to extract domain-slot pairs.

2.2. Progressive Domain-Slot Tracking

We define $B^{(t)} = \{b_1^{(t)}, b_2^{(t)}, ..., b_m^{(t)}\}$ as the belief state at the t-th turn, where $b_i^{(t)}$ is the i-th domain-slot pair with the label from $\{$`pointed`, `dontcare`, `none`$\}$. We also define the change from $B^{(t-1)}$ to $B^{(t)}$ as $\Delta^{(t)} = \{\delta_1^{(t)}, \delta_2^{(t)}, ..., \delta_m^{(t)}\}$, where $\delta_i^{(t)}$ is specified as follows:

$$\delta_i^{(t)} = \begin{cases} b_i^{(t)}, & b_i^{(t)} \neq b_i^{(t-1)} \\ \varnothing, & b_i^{(t)} = b_i^{(t-1)} \end{cases}. \tag{7}$$

Using $B_{ptd}^{(t)}$ to denote the `pointed` and `dontcare` states at the t-th turn of a dialogue, we define two turn-wise observations:

- If $|B_{ptd}^{(t-1)}| < |B_{ptd}^{(t)}|$ for most utterances, we call it `accumulating state triples`.
- If the changed states $|\Delta^{(t)}|$ is small, we say the dialogue has `adjacent state dependencies`.

The `accumulating state triples` phenomenon brings considerable difficulties to DST since there are increasingly more states to track. However, we can relieve this issue by the `adjacent state dependencies` notion. Since $|\Delta^{(t)}|$ is small in general, we propose to predict $B^{(t)}$ based on $B^{(t-1)}$ progressively rather than from scratch, enabling the multi-level attention module to focus more on the `change` between adjacent turns.

Concretely, we take the average of the progressive tracking output Y_{t-1}^{final} of the (t-1)-th turn and the multi-level attention output Y_t^G of the t-th turn. Since Y_{t-1}^{final} may not be correctly predicted, we apply a softmax layer as the smoothing operation to reduce the risk of error propagation:

$$Y_t^{final} = \frac{\text{Softmax}(Y_{t-1}^{final}) + Y_t^G}{2}. \tag{8}$$

For training, we employ cross entropy to calculate domain-level loss $L^D = \sum_{i=1}^{m} -\log(Y_i^D (y_i^{cls})^\intercal)$, slot-level loss $L^S = \sum_{i=1}^{m} -\log(Y_i^S (y_i^{cls})^\intercal)$ and global loss $L^G = \sum_{i=1}^{m} -\log(Y_i^{final} (y_i^{cls})^\intercal)$, where y_i^{cls} is the one-hot ground-truth label for the i-th domain-slot combination.

Finally, we use the final output to produce the final predictions. The overall loss L_{cls} for domain-slot prediction is the sum of these losses:

$$L_{cls} = L^D + L^S + L^G. \tag{9}$$

2.3. Shrunk Value Prediction

In the first stage, the domain-slot tracker generates a label for each domain-slot pair. While in the value prediction stage, we only predict values for those `pointed` domain-slot pairs. In this way, we significantly reduce the number of domain-slot candidates to be considered while leveraging the benefits of the generation-based approach.

Following previous works [7, 8], we employ a generator with the copy mechanism [14] to extract `value` words from both the dialogue context and open vocabulary. We use a bi-directional gated recurrent unit (BiGRU) as the dialogue context encoder and a single direction GRU as value decoder.

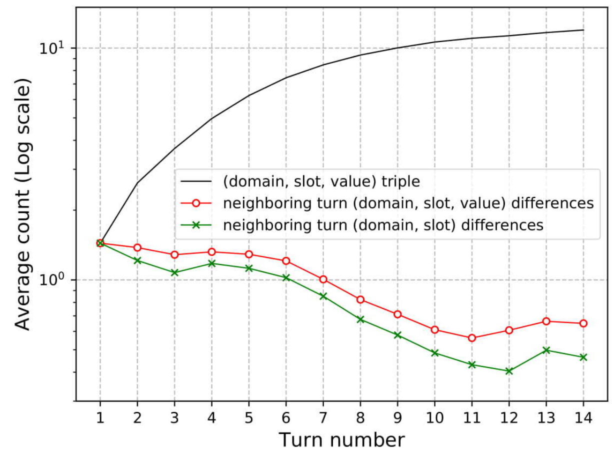

Fig. 3. Statistics on the MultiWOZ 2.1 training set. It shows the average count of dialogue state triples by different turns, as well as the counts of different domain-slot-value triples and different domain-slot pairs between two adjacent turns.

Initially, we feed the sum of the domain and slot embeddings to the decoder. At each decoding step j, we feed the decoder with the word embedding w_j and obtain its output hidden state h_j. We then calculate a dialogue utterance pointer $P_j^{\text{uttr}} \in \mathbb{R}^n$ and dialogue context vector $c_j \in \mathbb{R}^{d_h}$ by $P_j^{\text{uttr}} = \text{Softmax}(H_t h_j^\intercal)$ and $c_j = P_j^{\text{uttr}} H_t$, where $H_t \in \mathbb{R}^{n \times d_h}$ is the encoded context up to turn t.

Denote $E^V \in \mathbb{R}^{|V| \times d_e}$ as the embeddings of vocabulary words sizing $|V|$. We use both the hidden state h_j and the context vector c_j to attend to the vocabulary, leading to a vocabulary distribution, $P_j^{\text{vocab}} = \text{Softmax}(E^V(h_j + c_j))$. The final distribution $P_t^{\text{final}} \in \mathbb{R}^{|V|}$ is a weighted sum of P_t^{uttr} and P_t^{vocab}:

$$P_j^{\text{final}} = \alpha_j P_j^{\text{vocab}} + (1 - \alpha_j) P_j^{\text{uttr}}, \tag{10}$$

$$\alpha_j = \text{Sigmoid}(W_3[h_j; w_j; c_j] + b_3). \tag{11}$$

$W_3 \in \mathbb{R}^{1 \times 3d_h}$ and $b_3 \in \mathbb{R}^1$ are trainable parameters.

To train the value generator, we calculate the cross-entropy loss between P_{jk}^{final} and ground-truth values y^{gen}.

3. EXPERIMENTS

3.1. Datasets

MultiWOZ is a large task-oriented multi-domain dataset for dialogue state tracking. It has 10,438 dialogues and 11.06 turns on average for each dialogue. It involves 7 domains and 18 slots, which form 35 domain-slot pairs. We conducted evaluations on the latest MultiWOZ 2.1 dataset [15], which

Model	MultiWOZ 2.0 Acc. (%)	MultiWOZ 2.1 Acc. (%)
GLAD [2]	35.57	-
DST Reader [17]	39.41	36.4
COMER [9]	45.72	-
TRADE [8]	48.62	45.6
NADST [18]	50.52	49.0
SAS [19]	51.03	-
DST-SC [20]	**52.24**	49.6
PRO-DST (Ours)	51.48	**49.9**

Table 1. Overall results of our model and several baselines in joint goal accuracy. '-' means scores are not reported in the original papers.

fixes substantial errors in MultiWOZ 2.0 [16]. The results on MultiWOZ 2.0 are also reported for reference.

Statistics on the dialogue states of the MultiWOZ 2.1 training set are shown in Figure 3. Our findings are twofold. First, the number of domain-slot-value triples increases as the number of dialogue turns grows. Second, the differences between adjacent turns in domain-slot-value and domain-slot are both constantly small. These findings confirm our observations mentioned earlier.

3.2. Model Settings

Domain-Slot Tracker. We use the teacher forcing [21] to encourage the domain-slot tracker to focus on the most recent dialogue turns. During training, we use the ground-truth labels to replace the predicted Y_{t-1}^{final}. The early-stopping patience is 6. Besides, the utterances tokens are randomly masked off from the training input to enhance the generalization ability. We use Adam optimizer [22] with a 0.0001 learning rate to optimize the tracker. The batch size is 8 and the dropout ratio is 0.2. The word embeddings are initialized with 300 dimension GloVe embeddings [23] concatenated with 100 dimension character embeddings [24]. The hidden size d_h is equal to d_e = 400 for attention calculations.
Value Generator. The value generator has the same learning rate, dropout rate, optimizing, and early stopping settings as the domain-slot tracker. We use a fixed 0.5 teacher forcing ratio at each decoding step. For easier implementation, we set the batch size to 1 as in [9]. Another set of trainable word embeddings are adopted for value generation, also initialized with the GloVe embeddings concatenated with character embeddings.

3.3. Results and Ablation study

Following previous works, we use the joint goal accuracy and joint domain-slot accuracy metrics for evaluation. These metrics compare the predicted dialogue states and domain-slot pairs to the ground truth at each dialogue turn. As shown in Table 1, our model achieves competitive performance among the reported models on the MultiWOZ 2.0 and MultiWOZ 2.1 datasets. We also compare the time complexity with other methods as shown in Table 2. The analysis in performance and time complexity indicates our method achieves a balance in two perspectives, which is superior over the baselines.

Model	Best	Worst
COMER [9]	$\Omega(1)$	$O(n)$
TRADE [8]	$\Omega(n)$	$O(n)$
NADST [18]	$\Omega(1)$	$O(n)$
SAS [19]	$\Omega(n)$	$O(n)$
DST-SC [20]	$\Omega(n)$	$O(n)$
PRO-DST (Ours)	$\Omega(1)$	$O(n)$

Table 2. Time complexity of value sequence generation-based models. n represents the number of domain-slots.

Row	Model	Domain-Slot Acc.	Goal Acc.
1	Our Model	58.06	49.89
2	$-$ output of previous turn	56.28	48.4
3	$-$ smoothing of previous-turn output	51.14	44.75
4	$-$ domain&slot attention module	47.92	41.99
5	$-$ learnable domain&slot emb. $+$ fixed domain&slot emb.	58.06	49.80

Table 3. Results of ablation study on the MultiWOZ 2.1 dataset, where "$-$" and "$+$" mean removing or adding a component, respectively.

The ablation experiment results are shown in Table 3. The considerable performance drops displayed in row 2, row 3 and row 4 prove the effectiveness of our progressive tracker, smoothing strategy, and multi-level attention, respectively.

4. CONCLUSION

In this paper, we proposed a two-stage method for dialogue stage tracking, inspired by two important observations. We first model domain-slot pairs in a multi-level structure convenient for current turn to progressively incorporate the prediction of its previous turn. Then, the model generates values from both dialogue context and vocabulary for the predicted, rather than for all, domain-slot pairs. The progressive tracker is able to predict domain-slot pairs in parallel and reduce the number of domain-slot pairs for value prediction. Experimental results on MultiWOZ show that our method achieves competitive improvement over the considered baselines.

5. ACKNOWLEDGEMENT

This work was supported by the Fundamental Research Funds for the Central Universities (No.19lgpy220), and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355).

6. REFERENCES

[1] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young, "Neural belief tracker: Data-driven dialogue state tracking," in *ACL*, 2017.

[2] Victor Zhong, Caiming Xiong, and Richard Socher, "Global-locally self-attentive encoder for dialogue state tracking," in *ACL*, 2018.

[3] S. Liu, S. Liu, and W. Xu, "Gated attentive convolutional network dialogue state tracker," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6174–6178.

[4] J. Li, S. Zhu, and K. Yu, "A hierarchical tracker for multi-domain dialogue state tracking," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8014–8018.

[5] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," *arXiv preprint arXiv:1909.05855*, 2019.

[6] Guan-Lin Chao and Ian Lane, "BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer," in *Interspeech*, 2019.

[7] Puyang Xu and Qi Hu, "An end-to-end approach for handling unknown slot values in dialogue state tracking," in *ACL*, 2018.

[8] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung, "Transferable multi-domain state generator for task-oriented dialogue systems," in *ACL*, 2019.

[9] Liliang Ren, Jianmo Ni, and Julian McAuley, "Scalable and accurate dialogue state tracking via hierarchical sequence generation," in *EMNLP-IJCNLP*, 2019.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[11] Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim, "Sumbt: Slot-utterance matching for universal and scalable belief tracking," in *ACL*, 2019.

[12] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros, "Hierarchical multi-label classification networks," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80, pp. 5075–5084, PMLR.

[13] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Abigail See, Peter J Liu, and Christopher D Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*, 2017.

[15] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur, "Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines," *arXiv preprint arXiv:1907.01669*, 2019.

[16] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic, "multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," in *EMNLP*, 2018.

[17] Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur, "Dialog state tracking: A neural reading comprehension approach," in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 2019, pp. 264–273.

[18] Hung Le, Richard Socher, and Steven C.H. Hoi, "Non-autoregressive dialog state tracking," in *International Conference on Learning Representations*, 2020.

[19] Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu, "SAS: Dialogue state tracking via slot attention and slot information sharing," in *ACL*, July 2020.

[20] Yawen Ouyang, Moxin Chen, Xinyu Dai, Yinggong Zhao, Shujian Huang, and Jiajun Chen, "Dialogue state tracking with explicit slot connection modeling," in *ACL*, July 2020.

[21] Ronald J. Williams and David Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, June 1989.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[24] Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al., "A joint many-task model: Growing a neural network for multiple nlp tasks," in *EMNLP*, 2017.