Efficient Dialog State Tracking Using Gated-Intent based Slot Operation Prediction for On-device Dialog Systems

Pranamya Patil[1], Hyungtak Choi[2], Ranjan Samal[1], Gurpreet Kaur[1], Manisha Jhawar[1], Aniruddha Tammewar[1], Siddhartha Mukherjee[1]
[1] Samsung Research Institute, Bangalore
[2] Samsung Research, Samsung Electronics Co. Ltd., Seoul, Korea
{pran.patil, ht777.choi, ranjan.samal, k.gurpreet, m.jhawar, aniruddha.t, siddhartha.m}@samsung.com

Abstract

Conversational agents on smart devices need to be efficient concerning latency in responding, for enhanced user experience and real-time utility. This demands on-device processing (as on-device processing is quicker), which limits the availability of resources such as memory and processing. Most state-of-the-art Dialog State Tracking (DST) systems make use of large pre-trained language models that require high resource computation, typically available on high-end servers. Whereas, on-device systems are memory efficient, have reduced time/latency, preserve privacy, and don't rely on network. A recent approach tries to reduce the latency by splitting the task of slot prediction into two subtasks of State Operation Prediction (SOP) to select an action for each slot, and Slot Value Generation (SVG) responsible for producing values for the identified slots. SVG being computationally expensive, is performed only for a small subset of actions predicted in the SOP. Motivated from this optimization technique, we build a similar system and work on multi-task learning to achieve significant improvements in DST performance, while optimizing the resource consumption. We propose a quadruplet (Domain, Intent, Slot, and Slot Value) based DST, which significantly boosts the performance. We experiment with different techniques to fuse different layers of representations from intent and slot prediction tasks. We obtain the best joint accuracy of 53.3% on the publicly available MultiWOZ 2.2 dataset, using BERT-medium along with a gating mechanism. We also compare the cost efficiency of our system with other large models and find that our system is best suited for an on-device based production environment.

1 Introduction

With the rapid growth of internet and thus Internet of Things, smart devices including smartphones, TV, refrigerators, among others that can communicate with each other are being increasingly introduced in the market. Smart devices come with processing power, which opens up the capability of deploying AI solutions (Agarwal et al. 2020, Ghosh et al. 2021). These solutions also include on-device Conversational Agents (CA) and thus its components such as intent detection (Agarwal et al. 2021). These CAs such as Alexa, Bixby, and Google home, tend to be task-oriented, and perform the device-specific tasks.

A user of a smart device CA expects a quick action and response from the device, otherwise it's no better than manually performing the task. The low latency demands for on-device processing to reduce/remove network calls to a server. Even though the smart devices come with processing capabilities, usually the processing power and memory are very limited. This makes it very difficult to deploy large and complex DNN models on the device.

We are particularly interested in the task of Dialog State Tracking (DST), which is a crucial module of a CA. Many state-of-the-art (SOTA) DST systems, such as Zhao et al. (2021), Tian et al. (2021) are based on large language models, which need high processing power and memory during inference, and thus suitable for server side processing.

In this scenario, on-device systems can play a major role. They can operate on low resources and hence, can be run on mobile devices / edge processors. In addition to occupying lesser space and providing lower latency, they also require lesser RAM. They are better than server based

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 67 - 74

models with respect to privacy, security and non-reliability on network.

In light of these advantages, we focus on building high performance on-device DST. In this paper, we propose an efficient DST architecture, which can run in resource constrained environment and can provide comparable accuracy to other SOTA models on MultiWOZ 2.2 dataset (Zang et al., 2020) .

Majority of the open vocabulary based DST systems, predict/generate slot values at each turn. This is rather an inefficient approach for both latency and prediction accuracy. Kim et al. (2020), worked on solving this challenge and proposed Selective Overwriting Memory for efficient DST (abbreviated as SOM-DST), based on a two-step process consisting of State Operation Predictor (SOP) and Slot Value Generator (SVG) modules. SOP helps decide which slots' values need to be updated/generated, thus gating the amount of SVG requests made. As the two-step architecture achieves significant improvements in latency, we base our experiments on SOM-DST. In this work, we try to improve the SOP module, as the authors analyzed better possibility of improvements in SOP than SVG.

The SOM-DST system was trained on MultiWOZ 2.1 (Eric et al., 2019), which didn't have intent information. We work on MultiWOZ 2.2 dataset and make use of the intent annotation provided for each utterance, which may prove to be helpful for the SOP in a multi-task setting of intent and slot prediction. Intent in an utterance depicts the ulterior motive of the speaker. For example, intents for Restaurant domain are *find_restaurant* and *book_restaurant*, which represent the main motive of the speaker of finding/booking a restaurant. Intent information may help selecting an appropriate operation (SOP) for each slot. For example, if the conversation is about meeting at a restaurant for lunch, then a dialog turn carrying time information related to a different intent (such as "*We had been to the same restaurant yesterday at 4 PM*") needs to be eliminated for the SVG generation phase. We experiment with different strategies to fuse the information from different representation layers of intent and slot predictors.

SOM-DST makes use of BERT-base model, which is a large model, not suitable for on-device processing. In this work, we not only improve the performance with joint learning and different fusion techniques, but also reduce the model size by replacing BERT-base with the BERT-medium model, making the overall size of the model ~202 MB (binary PyTorch file), small enough to deploy on-device.

Our major contributions include:

1. We build a lightweight two-step DST system that can be deployed on-device, while providing competitive efficiency to the SOTA models.
2. We improve a previous two-step model (SOM-DST) efficiency by jointly predicting intent, domain, state operation and slot value generation.
3. We experiment with different fusion strategies such as self-attention and gating while concatenating representations at different levels, to achieve better multi-task performance

2 Related Work

SOTA: Most recent works which have achieved SOTA results on MultiWOZ 2.2 are based on large language models. Lee et al. (2021) in their work of using Schema-Driven Prompting for DST have used T5 language model (Raffel et al., 2020). Tian et al. (2021) have introduced a two-pass generation process in which the second pass amends the primitive dialog state which was generated from the first pass and alleviates unnecessary error propagation. They also use large language models: GPT-2 and PLATO-2, and the two-pass generation process would also increase the latency.

Rastogi et al. (2020), proposed a scalable DST architecture for Schema Guided Dataset (SGD) for task oriented virtual assistants which predicts intent along with slot values. Their baseline model consists of two modules: Schema Embedding Module which embeds the schema elements (intents, slots and categorical slot values) and State Update Module which predicts the active intent, requested slots, slot values and performs state update using utterance (current user turn and previous system turn) embeddings and schema embeddings.

Fusion: Fusion of information from intent prediction and previous belief state is performed using fusion method described in CrossViT (Chen et al, 2021). The major advantage of this technique is the patch based encoding using transformers and its fusion. In CrossViT (Chen et al, 2021), their main approach is to divide image into patches (preferably of different sizes) and to pass them through separate branches of transformer and to fuse these features. This approach gives better accuracy than many current CNN based SOTA

models for Image Classification Task in Computer Vision domain. Based on this paper, we got motivated to try different approaches to fuse information from intent prediction and previous belief state for efficient SOP module.

Intent logits information fusion with previous belief state is performed as explained in Meng et al. Meng et al have proposed the following:

i) Flexible contextual gazetteer representation (CGR) which is similar to gazetteer embedding but also has context and positional features.

ii) Mixture of Experts (MoE) - Gating for CGR and CWR (Contextual Word Representation) to selectively pass gazetteer and context info, so as to pass both syntactic as well as gazetteer info dynamically based on use case. They have used Joint CGR and CWR gating network to learn to balance contributions. This avoids feature overuse/underuse problem. We use the Mixture of Experts logic for fusing the information from intent and previous belief state.

3 Dataset

We use MultiWOZ 2.2 dataset. Following Wu et al. (2019), we use only five domains (restaurant, train, hotel, taxi, attraction) excluding hospital and police. Therefore, the number of domains is five, the number of slots is 30 and the number of intents is 12.

	Train	Test	Validation
#dialogs	8,420	999	1,000
#turns	54,981	7,368	7,374

Table 1: Statistics of MultiWOZ 2.2 dataset.

4 Baseline System (SOM-DST)

As discussed in the Section 1, in this work, we base our experiments on improving the performance and optimizing the cost of the SOM-DST architecture (depicted in Figure 1) by Kim et al. (2020). To improve the latency, the DST system is divided into two modules:

a. **State Operation Predictor (SOP)**: For each slot (defined in the ontology), classify it amongst a predefined set of labels (such as carryover, update, delete, don't-care). These label values help us identify which slot's value has to be generated/updated and which has to be modified, deleted, skipped etc. The input to the SOP module is formed by concatenating the current dialogue context with the previous belief state (slots and corresponding values). The input is passed through a BERT encoder to obtain encodings for each slot, which are further processed for operation classification.

b. **Slot Value Generator (SVG)**: This module generates value only for the slots in which update operation is predicted from SOP. SVG generates the slot values using a simple GRU based model.

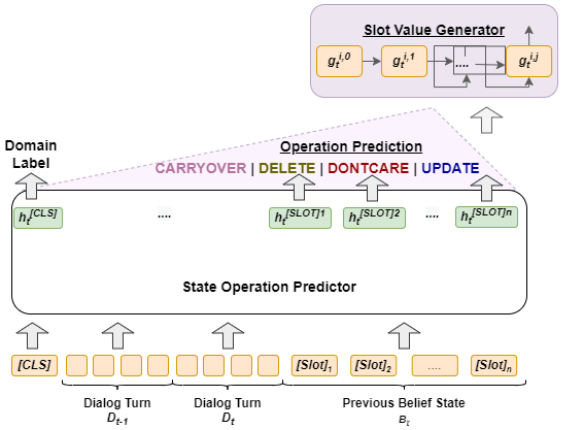

Figure 1: SOM-DST model architecture consisting of two sequential modules.

SOP gates the amount of SVG requests made. This is a very efficient way of determining the dialog state. In this work, we first replicate the results on MultiWOZ 2.2 dataset using the same architecture. We then experiment with different fusion experiments for the multitask learning of slot and intent predictions.

5 Fusion Experiments

We are mainly trying to fuse information from intent classification into State Operation Prediction. Introduction of the intent prediction into the SOM-DST architecture was designed in several ways as follows

5.1 Intent Prediction with Joint Loss Optimization

In SOM-DST, we are jointly optimizing loss from Domain Prediction, SOP and SVG modules. Domain prediction is done by adding classification head on BERT pooled output.

In this design (as depicted in Figure 2, experiment 1), we introduce Intent prediction as is done for Domain labels. The BERT-medium pooled output (represented by the *[CLS]* token) is passed through a linear layer of 512 x 12 (12 is possible number of intent labels, 512 is BERT-medium hidden dimension) to generate the intent

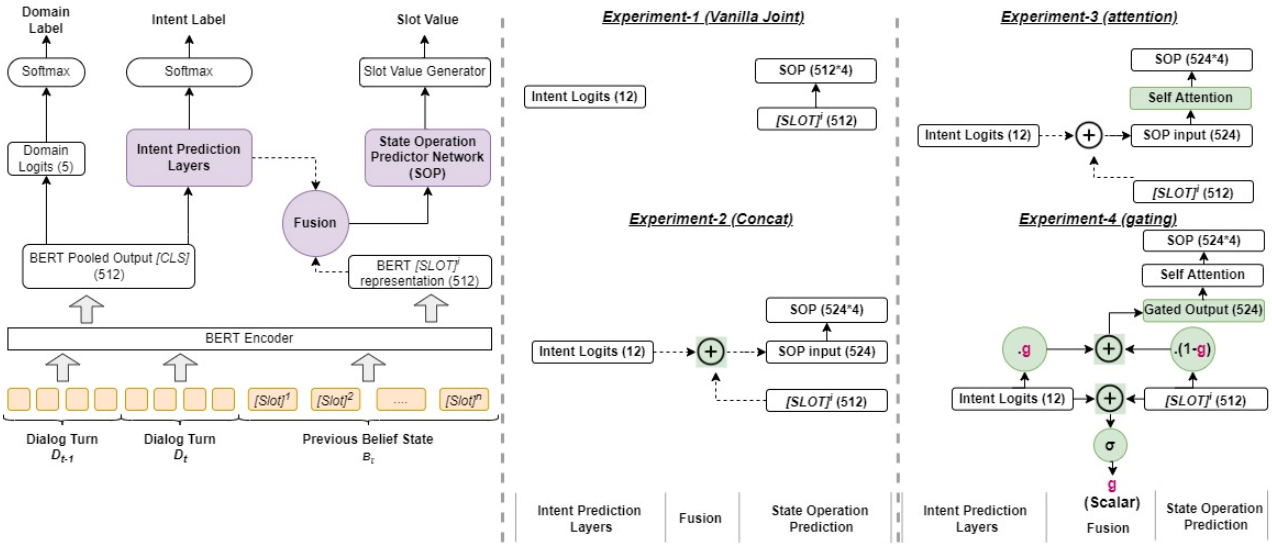

Figure 2: The leftmost part of the diagram shows the overall architecture of the multi-task learning including prediction of domain, intent, slot operation prediction and slot value generation. The entire context including previous and current dialog turns along with the previous belief state is passed through a BERT based encoder. The pooled embeddings ([CLS]) and SLOT embeddings are further used for prediction tasks. To improve performance of the slot prediction, we experiment with different strategies to infuse important information from different layers of the intent prediction network to that of the State Operation Prediction (SOP) module. The corresponding blocks are colored in purple. In the four experiments, we progressively add blocks and layers (marked with Green color). In Experiment-1 we try vanilla multi-task learning with joint loss optimization; later in Experiment-2 we concatenate intent logits with the SLOT logits for better access to the intent information in SOP; for better weighing of the concatenated SLOT and intent logits, we introduce a self-attention layer in Experiment-3; whereas in Experiment-4, using gating mechanism, we selectively infuse only the relevant intent information for more improvements.

logits. The model is jointly optimized along with intent using the joint cross-entropy loss. With this base model, we see a boost in the SOP classification results.

5.2 Concatenating intent logits and a layer from the SOP module

In conjunction with joint optimization, the intent logits are fed into the SOP module (via concatenating intent logits with BERT encoded "[SLOT]" tokens). This way we try to introduce the intent logits so that they have an impact on SOP. This is depicted in Figure 2, experiment 2.

5.3 Intent & Slot Self-Attention network

In this model architecture (depicted in Figure 2 experiment 3), we allow the intent logits to interact with the embedding inputs to SOP module (which are BERT encoded "[SLOT]" token from previous belief state input). This way the model can establish similarity between the previous belief state and intent in order to determine the SOP labels for current turn.

There are two approaches to generate similarity. First, the cross-attention way as mentioned in CrossViT (Chen et al, 2021). In this approach, the resultant cross attention matrix is large sized and is sparse. Moreover, the dimension of intent logits being far less than the belief state, the effect of intent gets nullified. Hence, we move on to an alternative way of self-attention (Vaswani et al, 2017). Here we concatenate the intent logits along with the previous belief state hidden representation and feed it through a single self-attention layer. By far this has been the best model to establish the similarity between the turn intent and previous belief state.

The cross attention technique (equation 1) can be represented as follows:

$$Q = W_q * x_1$$
$$K = W_k * x_2$$
$$V = W_v * x_2$$
$$S = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Where W_q, W_k, W_v are learnable parameters (weight matrices). x_1 is intent logits and x_2 is previous belief state's "[SLOT]" token embedding.

5.4 Gated-Intent Quadruplet State

5.4.1 Model Architecture

In all the design choices discussed before

section 5.4, we primarily mandated the use of intent logits or intent classification results in conjunction with the SOP input (which is BERT encoded "[SLOT]" token embedding). In case of topic steering or change in task-oriented discussions, we still force fit the non-related intent from the task to propagate into the model.

Yann et al. (2017), demonstrated a gated CNN network-based language model which was able to perform competitively against the large-scale recurrent models. Though gates were well known in recurrent networks, Yann et al. (2017), applied them to non-recurrent networks for the first time and the results were impressive. We adopted the same mechanism as Mixture of Experts from Meng et al. (2021) (depicted in Figure 2 experiment 4) and observed that the model was able to undo the adverse effect of force-fitting intent for the DST.

5.4.2 Intent Gating Mechanism

If $'X'$ represents the BERT Encoder pooled output of hidden state representation for the dialog turn and the previous belief state, $'W_i'$ represents the weight matrix for intent hidden layer, $'W'$ represents the weight matrix for the intent logits layer, then the output of the gating hidden layer (equation 2) is given as follows:

$$\begin{aligned} I_1 &= (X * W_i + c) \\ T_{concat} &= I_1 \oplus S_1 \\ g &= \sigma(T_{concat} * W + b) \end{aligned} \tag{2}$$

$$h_1(X) = (g * I_1) \oplus ((1 - g) * S_1) \tag{3}$$

Where $\oplus$ represents the concatenation operation, $'I_1'$ represents Intent Logits, $'S_1'$ represents BERT encoded "[SLOT]" tokens from previous belief state, $'g'$ represents gating value (generally a scalar), as expressed in equation 3.

The output from equation 3 is then passed through self-attention and then linear projection layer. Q is Query, K is Key, V is Value. W_q , W_k , W_v are learnable parameters (weight matrices).

$$\begin{aligned} Q &= W_q * h_1(X) \\ K &= W_k * h_1(X) \\ V &= W_v * h_1(X) \\ S &= softmax(\frac{QK^T}{\sqrt{d_k}})V \\ SOP_{OP} &= (S * W) + b \end{aligned} \tag{4}$$

Per slot predicted state operation is denoted by equation 4.

5.4.3 Loss Function

For the entire training, we have used the average cross entropy loss from each of the modules such as intent classification, domain classification, SOP and SVG module.

6 Experimental Setup

6.1 Evaluation Metrics

We use joint accuracy and F1 scores for different SOP modules for evaluating model performance. SOP labels classify each turn for each slot in one of the following categories

1. **CarryOver** - No change from the previous turn for that slot.
2. **Delete** - The previously entered slot value is cancelled/removed (set to none)
3. **Update** - Particular slot has to be updated with new value. Leads a call to SVG.
4. **Dontcare** - The slot value is not relevant and is set to "dontcare" literal.

6.2 Data Preparation

We have followed same preprocessing steps as in the case of Kim et al. (2020), with intent as an additional field extracted from MultiWOZ 2.2 data.

6.3 Training

We employ the pre-trained BERT-medium-uncased model for SOP and one GRU (Cho et al., 2014b) for SVG. The hidden size of the decoder and encoder is the same, which is 512. We use BertAdam as our optimizer (Kingma and Ba, 2015) and greedy decoding for SVG. The encoder of SOP makes use of a pre-trained model, whereas the decoder (GRU) of SVG needs to be trained from scratch. Therefore, we use different learning rate schemes for the encoder and the decoder. We use a batch size of 32 and set the dropout (Srivastava et al., 2014) rate to 0.1. We also utilize word dropout (Bowman et al., 2016) by randomly replacing the input tokens with the special [UNK] token.

The max sequence length for all inputs is fixed to 512. We train SOP and SVG jointly with early stopping and choose the model that reports the best performance (joint accuracy) on the validation set. We use teacher forcing 50% of the time to train the decoder. This is done so that the model is well accustomed to the test time scenario (i.e., intent from intent classifier output) and to intent from GT (so that, model doesn't face error propagation from intent prediction side).

We fuse the gated-intent logit features and the gated BERT encoded "[SLOT]" tokens following a mechanism similar to Mixture of Experts by Meng et al. (2021). We add two layers of Self-Attention and SOP classification head on top of the fused output for each slot. We train this model on Tesla GPUs.

7 Results

7.1 Joint Goal Accuracy (Overall Results)

We have achieved joint goal accuracy comparable to SOTA joint goal accuracy on MultiWOZ 2.2.

Model	Accuracy	Size
DS-DST (Zhang et al. (2019))	51.70	~440MB
SOM-DST baseline (Kim et al. (2020))	52	432MB
Gated-Self Attention DST (BERT-medium)	**53.30**	**202MB**
Gated-Self Attention DST (BERT-base)	**54.09**	**496MB**
Pegasus (Zhao et al. (2021))	56.60	>2.2GB
T5 (Zhao et al. (2021))	57.60	>891MB

Table 2: Joint Goal Accuracy on MultiWOZ 2.2

7.2 SOP Efficiency

F1 Score for State Operation Prediction (SOP) module.

Model	Operation			
	Delete	Update	Don't-Care	Carry-Over
SOM-DST (Baseline)	22.05	91.56	54.67	99.60
Intent Prediction (Joint Loss Optimization)	14.41	91.55	55.41	99.60
Appending intent logits-to SOP module	22.41	91.81	55.16	99.61
Intent and Slot-Self Attention	22.05	91.66	58.82	99.61
Gated-Intent (proposed model)	20.16	91.89	58.80	99.62

Table 3: SOP scores (F1) for each operation for dialog state borrowed from GT

8 Analysis

In Table 2, we compare our system's performance with other important works. Our Gated-Self Attention based model achieves a Joint Goal Accuracy (JGA) of 54.09 using BERT-base, and of 53.30 using BERT-medium. The system performance is comparable with the current SOTA results, while also providing the benefit of lesser processing. We achieve a performance improvement of 1.3% JGA, over the SOM-DST baseline.

We also compare the sizes of the different pre-trained models used by different systems, which gives a hint of the comparative memory efficiency of the models. Compared to T5 and Pegasus, our model makes use of BERT-medium , which is 4 times and 10 times smaller, respectively. Our model size is 202 MB which makes it feasible to deploy on-device.

As presented in Table 3, we also observe significant improvements in SOP efficiency, indicating optimization of the calls made to the time-consuming Slot Value Generator (SVG) module, further decreasing the overall latency of the system. The improvements are consistent across all the state-operations (Delete, Update, Don't Care, and Carry Over).

8.1 Future scope of enhancements

Similar improvisation can further be extended to dialog acts, which are more generic than intents, for SOP tasks.

We also plan to explore quantization techniques for reducing the model size without affecting the prediction results.

Another technique that has shown benefits in the task of named entity recognition is the use of external knowledge bases, for ever-expanding dynamic entities. We can further improve our system by incorporating such knowledge.

A limitation of our current system is the upper cap on the length of input (512 tokens). We would like to explore techniques to handle longer input sequences.

9 Conclusion

From our experiment results, we can conclude that using gating based self-attention on the intent logits for state operation prediction improves the accuracy. There is also a significant reduction in model size and latency when compared to other

existing SOTA models which use large pre-trained language models. This makes our model more suitable for on-device based production environment.

References

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 567–582, Online. Association for Computational Linguistics.

Rastogi, A., Zang, X., Sunkara, S., Gupta, R., & Khaitan, P. (2020). Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 8689-8696. https://doi.org/10.1609/aaai.v34i05.6394

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pages 109–117, Online. Association for Computational Linguistics.

Yann N. Dauphin, Angela Fan, Michael Auli, David Grangier. 2017. Language Modeling with Gated Convolutional Networks. Proceedings of the 34th International Conference on Machine Learning, in Proceedings of Machine Learning Research 70:933-941 Available from https://proceedings.mlr.press/v70/dauphin17a.html

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. *Adam: A method for stochastic optimization. In ICLR*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective Sequence-to-Sequence Dialogue State Tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue State Tracking with a Language Model using Schema-Driven Prompting. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable Generation for Dialogue State Tracking. In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pages 80–92, Online. Association for Computational Linguistics.

Chen, Chun-Fu Richard, Quanfu Fan, and Rameswar Panda. "Crossvit: Cross-attention multi-scale vision transformer for image classification." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 357-366. 2021.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).

BERT-Medium: https://huggingface.co/google/bert_uncased_L-8_H-512_A-8

Meng, Tao, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. "GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input." In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

V. Agarwal, S. D. Shivnikar, S. Ghosh, H. Arora and Y. Saini, "LIDSNet: A Lightweight on-device Intent Detection model using Deep Siamese Network,"

2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1112-1117, doi: 10.1109/ICMLA52953.2021.00182.

S. Ghosh, S. V. Gothe, C. Sanchi, and B. R. K. Raja, “edATLAS: An Efficient Disambiguation Algorithm for Texting in Languages with Abugida Scripts,” in 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Jan 2021, pp. 325–332.

V. Agarwal, S. Ghosh, K. Ch, B. Challa, S. Kumari, Harshavardhana, and B. R. Kandur Raja, “EmpLite: A lightweight sequence labeling model for emphasis selection of short texts,” in Proceedings of the Workshop on Joint NLP Modelling for Conversational AI @ ICON 2020. Patna, India: NLP Association of India (NLPAI), Dec. 2020, pp. 19–26.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani Tür. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines.

Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." J. Mach. Learn. Res. 21.140 (2020): 1-67.

Zhang, J.G., Hashimoto, K., Wu, C.S., Wan, Y., Yu, P.S., Socher, R. and Xiong, C., 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.