Attention Modulation for Zero-Shot Cross-Domain Dialogue State Tracking

Mathilde Veron[1,2] and **Guillaume Bernard**[2] and **Olivier Galibert**[2] and **Sophie Rosset**[1]

[1]Université Paris-Saclay CNRS, LISN, Orsay, France; [2]LNE, Trappes, France;

[1]{name.lastname}@lisn.fr; [2]{name.lastname}@lne.fr

Abstract

Dialog state tracking (DST) is a core step for task-oriented dialogue systems aiming to track the user's current goal during a dialogue. Recently a special focus has been put on applying existing DST models to new domains, in other words performing zero-shot cross-domain transfer. While recent state-of-the-art models leverage large pre-trained language models, no work has been made on understanding and improving the results of first-developed zero-shot models like SUMBT. In this paper, we thus propose to improve SUMBT zero-shot results on MultiWOZ by using attention modulation during inference. This method improves SUMBT zero-shot results significantly on two domains and does not worsen the initial performance with the significant advantage of needing no additional training.

1 Introduction

Task-oriented dialogue systems aim to provide information and perform tasks requested by a user during a dialogue (*e.g.*, booking a train ticket or finding a restaurant). As the dialogue progresses, the user may add some criteria or change its goal, so the system needs to track the current goal of the user at each dialogue turn for the dialogue to succeed. The associated task is called Dialogue State Tracking (DST) and consists, in its most studied form, in updating the slots mentioned by the user (see Figure 1). State-of-the-art models rely on deep learning models. However, a highly desirable feature of dialogue systems is the ability to scale to new domains without retraining but by taking advantage of knowledge already acquired in previous domains. Thus in this paper we study "leave-one-out" cross-domain zero-shot transfer. For each domain, a model is trained on dialogues that do not contain slots of the target domain and is then evaluated on dialogues containing slots of the target domain.

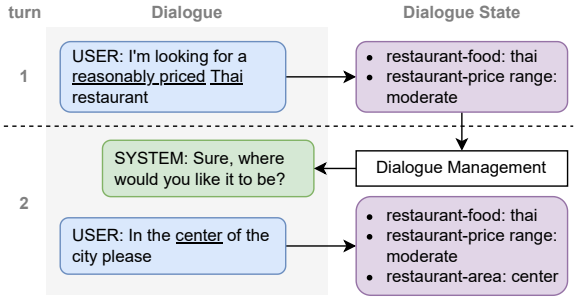


Figure 1: Example of dialogue along with the dialogue state at each turn.

Zero-shot cross-domain transfer studies on DST are relatively recent and are mainly conducted on the MultiWOZ dataset (Budzianowski et al., 2018)[1]. Such zero-shot learning was first applied to TRADE and SUMBT models (Campagna et al., 2020), where TRADE (Wu et al., 2019) relies on an RNN and SUMBT (Lee et al., 2019) on the pre-trained language model BERT (Devlin et al., 2019) and an RNN. Instead of building new architectures, recent state-of-the-art models leverage large generative pre-trained language models like GPT-2 (Radford et al., 2019) or T5 (Raffel et al., 2020), and work on the form of the input itself by incorporating slot descriptions (Lin et al., 2021b; Zhao et al., 2022), showing labeled examples (Gupta et al., 2022), or considering a slot as a question (Li et al., 2021; Lin et al., 2021a).

However, no further work has been conducted on understanding and improving the results of first-developed models. Thus in this paper we propose different architectural variants of SUMBT and introduce attention modulation to improve cross-domain zero-shot results on MultiWOZ 2.0.

[1]Schema-Guided Dialogue dataset (Rastogi et al., 2020) is also used but distinguishes only seen and unseen data, and thus does not allow cross-domain transfer analysis.

Proceedings of 3rd Workshop on Computational Approaches to Discourse, pages 86–91
October 16 – 17, 2022.

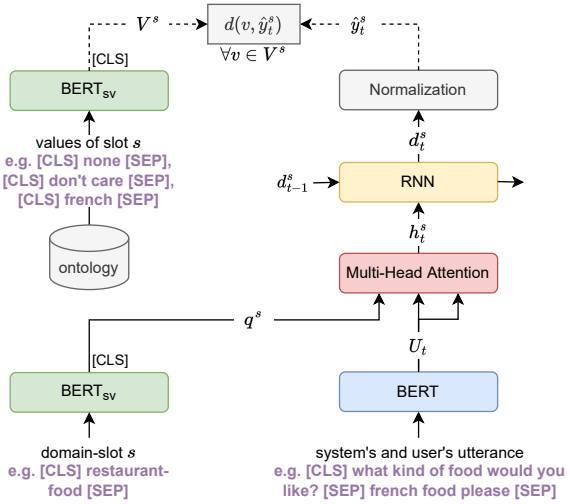


Figure 2: Architecture of SUMBT (Lee et al., 2019)

2 SUMBT

The main idea of SUMBT is to match each slot-name to a slot-value from an ontology given a dialogue turn (a system's and a user's utterance). The architecture of the model is illustrated in Figure 2[2]. During inference, any domain/slot-name pair can be used as query input as long as the ontology contains the list of values associated with the domain/slot-name pair. Trained SUMBT models can thus be applied to new domains after updating the ontology, and the models can predict new slots never seen during training.

We re-implemented our own version of SUMBT and conducted zero-shot cross-domain experiments. Transfer is measured by computing the Joint Goal Accuracy (JGA) only on the slots of the target domain. It consists of the percentage of turns from all dialogues where all targeted slots-names are associated with the correct slot-value. All experiments are run on 5 random seeds. In the first line of Table 1, we can observe that SUMBT performs poorly even if its ontology is updated before testing with the slot-value list of each slot-name from the target domain. Looking more closely at the model's predictions, we notice that SUMBT generally tends to predict the slot-value `none` more than it should. In fact, the proportion of `none` values in training data is 71%, while the model predicts 78% of the times the value `none` on test data of the domains used during training. When applying the model to unknown domains, the proportion increases on average to 88% and can even get to 99% in the case of the attraction domain. It shows that this tendency intensifies when a new slot never seen during training is queried.

[2]See Appendix A for further information.

3 Attention Modulation

Motivated by previous observations, we propose a method called *attention modulation* to push the model to predict the slot-value `none` less frequently for unknown slots. Specifically, this would apply when predicting the dialogue state of a dialogue turn that refers to an unknown domain. However, doing this could lead the model to predict any other value except the correct one. Thus we also describe two variants of SUMBT, aiming to take advantage of similarities that naturally exist between the slots of the different domains. We hypothesize that it would help the model to increase transfer between domains and that our method would be more effective on these variants.

3.1 Method

SUMBT relies on a multi-head attention layer, which basically repeats the Scaled Dot-Product Attention multiple times (Vaswani et al., 2017)[3]. This layer enables the model to draw its attention to tokens related to the queried slot. The attention mechanism takes as input three matrices: Q a set of queries, K a set of keys, and V a set of values. In our case, we have $Q \in \mathbb{R}^{1\times d}$, where Q corresponds to q^s the domain/slot-name pair encoded by $BERT_{sv}$ and d denotes the dimension of the BERT model. $K \in \mathbb{R}^{sl\times d}$ and $V \in \mathbb{R}^{sl\times d}$ both correspond to the concatenation of a system's and a user utterance (a dialogue turn) encoded by $BERT$ also noted $U_t = \{u_{t,i}\}_{i\in[\![0,sl]\!]}$, where t denotes a unique turn index over all dialogues and sl the maximum number of tokens that can be encoded by $BERT$ including the special tokens `[CLS]` and `[SEP]`. The attention mechanism is formalized as follow:

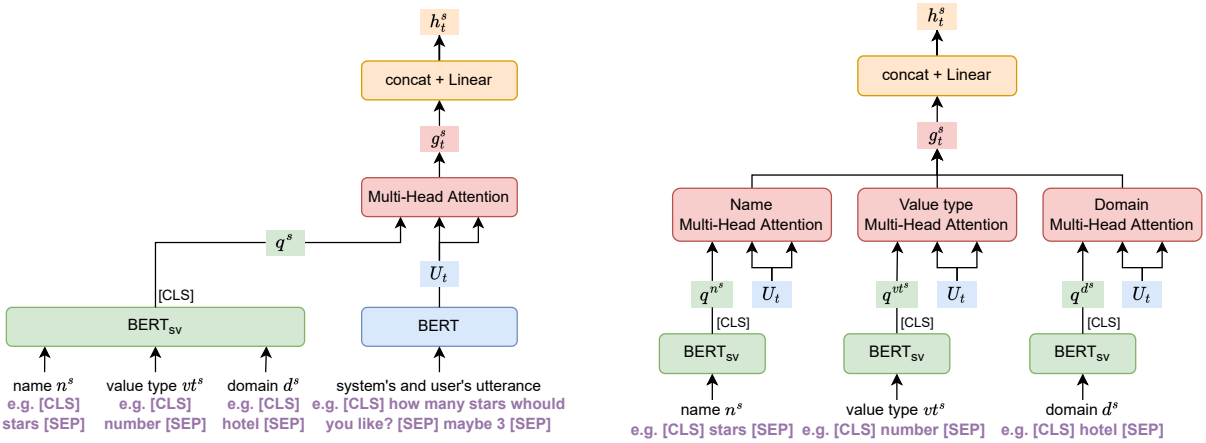
$$Attention(Q, K, V) = (w_{t,i}^{d^s}) \cdot V \quad (1)$$

$$\text{with } (w_{t,i}^{d^s})_{i\in[\![0,sl]\!]} = softmax\left(\frac{s_{t,i}^{d^s}}{\sqrt{d}}\right) \quad (2)$$

$$\text{and } (s_{t,i}^{d^s})_{i\in[\![0,sl]\!]} = Q \cdot K^T \quad (3)$$

Where d^s denotes the domain associated to the slot s and $w_{t,i}^{d^s}$ corresponds to the attention weights applied to U_t (the values matrix V) after normalizing the attention scores $s_{t,i}^{d^s}$.

[3]For illustration purposes in this paper, the dimensions do not take into account the number of heads.



(a) triple query

(b) triple attention

Figure 3: Variants of SUMBT. After h_t^s the architecture remains the same than the original model in Figure 2.

In their paper, SUMBT authors found out that the attention weights were high on the special tokens `[CLS]` and `[SEP]` when the slot-value `none` was predicted. To push the model to predict values other than the value `none`, we can then simply reduce the attention weights on these special tokens. We call this method *attention modulation* and defined it as follow:

$$(w_{t,i}^{d^s})_{i \in [\![0,sl]\!]} = softmax\left(\frac{\alpha_{t,i}^{d^s} \cdot s_{t,i}^{d^s}}{\sqrt{d}}\right) \quad (4)$$

$$\text{with } \alpha_{t,i}^{d^s} = \begin{cases} 0 & \text{if } d^s \in ND \text{ and } u_{t,i} \in ST, \\ 1 & \text{otherwise.} \end{cases} \quad (5)$$

Where ND is the set of new domains never seen during training, and ST is the set of special tokens `[CLS]` and `[SEP]`. This method is simple yet attractive since it does not need any additional training and can be directly applied to the model during inference.

3.2 Model Variants

Regarding SUMBT zero-shot results, it seems that it is not able to take advantage of the similarities that exist between each domain. In fact, some slots can share the same name, the same type of values, or even the same values. To boost cross-domain transfer, we describe each slot with its domain, its name, and the type of its values following (Lin et al., 2021b) "slot type" descriptions. We suppose that variants of SUMBT incorporating these descriptions should benefit more from attention modulation than the original model. We thus propose two main variants of SUMBT:

- **With triple query** (Figure 3a): The query q^s consists here in a matrix of 3 vectors corresponding to the name, the type of values, and the domain of the queried slot, the three being encoded by $BERT_{sv}$. Since we now have $q^s \in \mathbb{R}^{3 \times d}$, the multi-head attention layer outputs $g_t^s \in \mathbb{R}^{3 \times d}$. We thus reshape the output by concatenating the three vectors and by using a linear layer $h_t^s = g_t^s W + b$ with $W \in \mathbb{R}^{3*d \times d}$ followed by ReLU activation (Nair and Hinton, 2010).
- **With triple attention** (Figure 3b): We use 3 independent multi-head attention layers and input respectively the name, the type of values, and the domain of the queried slot, the three being encoded by $BERT_{sv}$. The outputs of each multi-head attention layer is then concatenated, and the resulting vector is reshaped the same way as previously. We suppose the independent training to favor more transfer.

For these two variants, as well as the original SUMBT model, we also add variants where the weights of the utterance encoder $BERT$ are fixed during training. We suppose this could help the model to generalize to unknown domains. Fixing its weights also has the advantage of reducing the computation cost per epoch considerably.

3.3 Experiments and Results

In these experiments, we used an oracle to detect the domain associated to the dialogue turn. The attention modulation is applied only on the query or the attention layer related to the domain, respectively for the *triple query* and the *triple attention*

Version	Modulation	Attraction	Hotel	Restaurant	Taxi	Train
Original	none	23.57 ±0.86	14.51 ±1.23	17.19 ±0.84	60.41 ±0.12	21.31 ±0.91
	on slot attn.	25.03 ±3.04	14.23 ±1.07	17.81 ±1.07	**60.48** ±0.15	21.25 ±0.88
		+1.46 ±2.19	−0.28 ±0.24	+0.62 ±1.46	+0.08 ±0.11	−0.06 ±0.05
+ frozen BERT	none	23.29 ±0.25	15.09 ±0.31	14.94 ±1.26	60.29 ±0.17	22.61 ±0.18
	on slot attn.	28.00 ±1.06	15.62 ±0.48	17.30 ±0.88	60.28 ±0.17	22.62 ±0.19
		+4.71 ±1.02	+0.53 ±0.22	+2.36 ±1.12	−0.01 ±0.03	+0.01 ±0.02
w/ triple query	none	23.56 ±2.09	16.02 ±1.17	18.16 ±1.19	56.11 ±3.60	21.42 ±1.59
	on domain query	25.40 ±1.78	16.14 ±0.95	**19.13** ±0.80	56.26 ±3.71	21.43 ±1.62
		+1.85 ±2.88	+0.12 ±0.41	+0.97 ±0.64	+0.15 ±0.26	+0.01 ±0.04
+ frozen BERT	none	24.52 ±1.07	15.92 ±0.78	15.58 ±0.32	58.17 ±1.75	22.61 ±0.33
	on domain query	25.58 ±1.36	15.90 ±0.70	16.99 ±0.58	58.13 ±1.77	22.63 ±0.31
		+1.06 ±1.23	−0.02 ±0.46	+1.40 ±0.68	−0.04 ±0.10	+0.02 ±0.03
w/ triple attn.	none	23.70 ±4.51	16.06 ±0.90	16.41 ±2.46	56.88 ±3.31	22.54 ±0.32
	on domain attn.	28.53 ±4.99	16.37 ±0.88	18.29 ±2.00	56.96 ±3.38	22.58 ±0.33
		+4.83 ±2.42	+0.31 ±0.09	+1.88 ±1.63	+0.08 ±0.08	+0.04 ±0.05
+ frozen BERT	none	23.32 ±1.64	15.55 ±0.90	15.65 ±1.20	59.68 ±0.83	**22.74** ±0.07
	on domain attn.	**29.83** ±1.57	**17.09** ±1.37	16.80 ±1.30	59.72 ±0.84	**22.74** ±0.07
		+6.51 ±0.87	+1.54 ±0.68	+1.15 ±0.60	+0.04 ±0.06	−0.00 ±0.03

Table 1: JGA of different variants of SUMBT on MultiWOZ 2.0 cross-domain zero-shot experiments with and without modulation. The columns denote the target domain and the ± sign denotes the standard deviation.

variant. The results are shown in Table 1. First, if we look at the results without modulation, it seems that the proposed variants do not increase cross-domain transfer in a general manner. On the attraction domain, the results of the different variants are similar to the SUMBT original ones. On the hotel and train domains, all variants perform better than the original. However, on the restaurant and taxi domains, almost all variants perform worst than the original, except the *triple query* variant on the restaurant domain. We also observe that fixing BERT weights during training does help the variant around half of the time to perform better than when fine-tuning BERT, so we cannot state that it is beneficial for transfer. Note that overall, fixing BERT weights gives less variation in the results.

Now, when looking at the results with modulation, we observe that the variant *triple attention* with a frozen BERT and modulation gets the overall best results on the attraction and the hotel domain with respectively a high increase of 6.26 and 2.58 points compared to SUMBT original without modulation. On the restaurant domain, the variant *triple query* with a fine-tuned BERT and modulation gets the best results with an increase of 1.94 points compared to SUMBT original without modulation. However, modulation does not seem to impact the taxi and train domains.

In order to better observe the actual benefit of modulation, we compute for each model trained on a specific seed the difference in its performance with and without modulation. The resulting differences are averaged across variants and domains and correspond to the third line of each variant in Table 1. In a general manner, we can see that modulation increases performance. In fact, the difference is almost always positive, and if not, it is contained in the standard deviation or close to it. On the attraction and hotel domains, the *triple attention* variants benefit more from modulation than the *triple query* ones. This suggests that the fact that the name, the type of values, and the domain of the queried slot have their own attention mechanism is more beneficial for transfer. More precisely, on these two domains the variant *triple attention* with a frozen BERT is the one that benefits the most from modulation with an increase of respectively +6.51 and +1.54. Surprisingly, modulation seems to work fine on SUMBT original with a frozen BERT on the attraction and restaurant domains. On the restaurant domain, the *triple query* and *triple attention* variants seem to benefit similarly from modulation. However, on taxi and train domains, modulation has a negligible impact on the performance of all variants. Apart from these two domains, modulation seems to have a better impact when BERT is frozen (two-thirds of the time).

4 Conclusion and Future Work

In this paper we proposed different variants of SUMBT and introduced *attention modulation*. This method successfully improves SUMBT original

cross-domain zero-shot results on the attraction and the hotel domains by respectively 6.26 and 2.58 points with the *triple attention* variant, while not needing any additional training and never worsening original results. For further work, we plan to analyze in detail the results and conduct additional experiments to understand better the impact of attention modulation on the different domains. For example, we plan to introduce a variable β in place of the value 0 in equation 5 to study how changing the value of β can affect evaluation results with modulation. We also plan to study the possibility of extending the attention modulation to other architectures.

Reproducible Research

In the spirit of reproducible research, we release our code as open source available at https://github.com/mathilde-veron/attention-modulation-zero-dst.

Acknowledgements

This work has been funded by French ANRT under CIFRE PhD contract # 2019/0628. It was also possible thanks to the Saclay-IA computing platform and was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011012609R1).

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Yuan Cao, Abhinav Rastogi, and Yonghui Wu. 2022. Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4541–4549, Seattle, United States. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074, Online. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021a. Zero-shot dialogue state tracking via cross-task transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021b. 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*, pages 5640–5648, Online. Association for Computational Linguistics.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The

schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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 (Volume 1: Long Papers)*. Association for Computational Linguistics.

Jeffrey Zhao, Raghav Gupta, Yuanbin Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *ArXiv*, abs/2201.08904.

A SUMBT

We describe here the Slot-Utterance Matching Belief Tracker (SUMBT) (Lee et al., 2019) architecture as well as the way it is trained and how it works during inference . The main idea of SUMBT is to match each slot-name to a slot-value from an ontology given a dialogue turn (a system's and a user's utterance). The architecture of the model is illustrated in Figure 2.

The text corresponding to the domain/slot-name pair is first encoded by a BERT model (Devlin et al., 2019) $BERT_{sv}$ and the output of the `[CLS]` token is retrieved to obtain a overall representation of the domain/slot-name pair as a vector q^s. The text corresponding to the system's and the user's utterance is also encoded by a BERT model $BERT$ so that each token of the utterance are represented of contextual vectors, resulting in the matrix U_t. Note that the utterance encoder $BERT$ is fine-tuned during training but that the weights of $BERT_{sv}$ are fixed . The encoded domain/slot-name pair is then used as query in the multi-head attention layer and the encoded utterances as key and value. It enables the model to draw its attention to the tokens that are related to the queried slot and outputs an overall representation of these tokens. Since DST is about updating the current state of the dialogue, the model needs information about the past state of the dialogue. This is performed thanks to the RNN. Finally, each slot-value from the ontology corresponding to the queried slot is encoded by $BERT_{sv}$, resulting in a matrix V^s, and the euclidean distance between each vector v of V^s and the normalized output of the RNN $\hat{y}_t^s$ is computed.

During training, the model learns to minimize the distance between $\hat{y}_t^s$ and y_t^s the vector of the target slot-value of the queried slot and to maximize the distance with the other slot-values vectors $v \neq y_t^s$ by using the cross-entropy loss. During inference, the predicted slot-value for the queried slot consists in the slot-value which gives the smallest distance to $\hat{y}_t^s$.