Towards end-to-end multi-domain dialogue modelling

Technical Report
CUED/F-INFENG/TR.706

29th September 2018

Paweł Budzianowski, Iñigo Casanueva,
Bo-Hsiang Tseng, Milica Gašić

{pfb30,mg436}@cam.ac.uk

Department of Engineering, Cambridge University

ISSN: 0951-9211

1 Introduction

Statistical approaches to dialogue modelling allow automatic optimisation of the Spoken Dialogue Systems (SDS) [Young, 2002]. SDS typically comprise various statistical components. This includes a spoken language understanding module, which takes a sentence as input and gives a dialogue act as output, a dialogue belief state tracker that predicts user intent and track the dialogue history, a dialogue policy to determine the dialogue flow, and a natural language generator to convert conceptual representations into system responses.

At the other end of the spectrum, sequence to sequence learning [Sutskever et al., 2014] has inspired several efforts to build end-to-end trainable conversational systems [Serban et al., 2015, Wen et al., 2017b]. This family of approaches treats dialogue as a source to target sequence transduction problem, applying an encoder network [Cho et al., 2014] to encode a user query into a distributed vector representing its semantics, which then conditions a decoder network to generate each system response. These models are limited to one domain and involve the pre-training of some of their components. Moreover, they typically require a large amount of data to train.

In this report we introduce a multi-domain dialogue architecture with cross-domain database pointer and the attention over the input sequence. The model is inspired by the previous works of [Wen et al., 2017b, Eric et al., 2017]. The model is evaluated on a recently collected multi-domain dataset spanning across many domains. Initial benchmarks on dialogue-context-to-text generation are provided illustrating a room for improvement. We discuss inefficiencies of the current architecture and possible future work.

2 End-to-end dialogue modelling

Advances in sequence to sequence learning [Sutskever et al., 2014] has inspired several efforts to build end-to-end trainable, task-oriented conversational systems where modular approach [Young et al., 2013b] can be replaced with one neural architecture. This approach should in theory reduce the cost of building and maintaining task-oriented systems and ease-up transfer learning between different domains.

One family of such approaches treats dialogue as a source to target sequence transduction problem, applying an encoder network [Bahdanau et al., 2014] to encode a user query into a distributed vector representing its semantics, which then conditions a decoder network to generate each system response. [Eric et al., 2017] proposed to combine attentions over the input sequence and the database. The output probabilities from both sources are combined to form the generated answer. Such architectures do not break differentiability, however, these models typically require a large amount of data to train [Bordes et al., 2017, Eric et al., 2017].

[Wen et al., 2017b] proposed a neural network-based model for task-oriented dialogue systems by balancing the strengths and the weaknesses of the two research communities. The model is end-to-end trainable but modularly connected; it does not directly model the user goal, but nevertheless, it still learns to accomplish the required task by providing relevant and appropriate responses at each turn. The model consists of a separate belief tracker, a database pointer

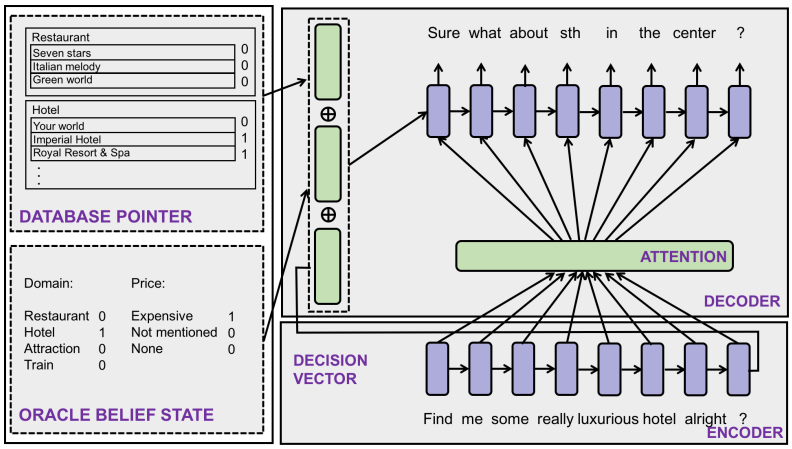

Figure 1: An architecture of the multi-domain dialogue system.

which is based on results from the understanding, and policy and generation modules.

All of these models were created and evaluated on simple and single-domain oriented datasets. Most of dialogues have around 3 system turns making this more similar to a question-answering set-up. Moreover, reinforcement learning was used only as a fine-tuning phase to improve the performance per-turn level.

3 Dialogue-Context-to-Text Generation

In this section we focus on modelling the dialogue management and response generation components. These problems can either be addressed separately [Young et al., 2013a], or jointly in an end-to-end fashion [Bordes et al., 2017, Wen et al., 2017b, Li et al., 2017]. We experimented with a baseline neural response generation model with an *oracle* belief-state obtained from the wizard annotations. This allows us to obtain a clear benchmark where the performance of the composite of dialogue management and response generation is completely independent of the belief tracking. Following [Wen et al., 2017b] which frames the dialogue as a context to response mapping problem, a sequence-to-sequence model [Sutskever et al., 2014] is augmented with an oracle belief tracker and a discrete database accessing component as additional features to inform the word decisions in the decoder. Figure 1 presents the architecture of the system.

3.1 Model

3.1.1 Intent modelling

We consider a dialogue that consists of T turns. Where it does not create confusion, we denote by L a length of both user and system sentences per turn. The full ontology consists of D domains, each with S_d slots and V_d values. We differentiate between a user output (u) and a machine (system) output (m). At each turn, the encoder takes a sequence of input tokens $u_t = (w_0^t, w_1^t, ..., w_L^t)$ and uses a (bi-directional) recurrent neural network to output a distributed user

utterance representation $\mathbf{u}_t$ which is the final hidden state:

$$\mathbf{u}_t = \mathbf{h}_L^u = \mathrm{RNN}_\theta(u_t).$$

[Wen et al., 2017b] defines this as a distributed intent representation which replaces the hand-coded dialogue act representation.

The summarised belief state is formed by going through the user belief state. For each slot, we encode whether the slot was not mentioned until this turn, some value has been given or the user does not care about this constraint. This gives 3-bin one-hot encoding for each slot. The global belief state vector is then formed by concatenating all slot-dependent vectors over all domains:

$$\mathbf{b}_t = \bigoplus_d \bigoplus_s \mathbf{b}_{s_d}.$$

Given an oracle belief state $\mathbf{b}_t$, we can use this information to query the database for all the domains. Based on the numbers of entities in the database that satisfy the query, we form a 6-bin one-hot encodings for each domain $\mathbf{kb}_d$. The vector indicates if 0, 1, 2, 3, 4 or more entities were matched by the query. All domain-dependent vectors are then concatenated:

$$\mathbf{kb}_t = \bigoplus_d \mathbf{kb}_d$$

to form a global domain vector.

3.1.2 Decision making

In the next step, a policy vector is created to mimic the dialogue manager in the traditional modular approach. The intent vector $\mathbf{u}_t$, the belief state vector $\mathbf{b}_t$ and the knowledge database vector $\mathbf{kb}_t$ are combined together and processed through a nonlinear layer [Wen et al., 2017b]:

$$\mathbf{a}_t = \tanh(\mathbf{W}_u\mathbf{u}_t + \mathbf{W}_k\mathbf{kb}_t + \mathbf{W}_b\mathbf{b}_t).$$

This vector can be seen as a continuous version of a system act in the traditional modular approach summarizing the current state and action in a high-dimensional space.

3.2 Generation

The generation module uses a language model in the form of the recurrent neural network that outputs probabilities over the vocabulary set at each time step:

$$P(w_{j+1}|w_j, \mathbf{h}_{j-1}^m) = \mathrm{softmax}(\mathrm{RNN}(w_j, \mathbf{h}_{j-1}^m)),$$

where w_j is the last output token and $\mathbf{h}_{t-1}^m$ is the hidden vector from the previous step. We condition a language generator through the action vector $\mathbf{a}_t$ by using it as the first hidden vector, i.e:

$$\mathbf{h}_0^m = \mathbf{a}_t.$$

To begin the generation process we use a special token (signifying the beginning of a sentence, SOS) as w_0. The generation process stops when the network outputs a special token informing about the end of the sentence (EOS).

The standard cross entropy is adopted as our objective function to train a language model:

$$L(\theta) = \sum_t \sum_j y_j^t \log p_j^t,$$

where y_j^t and p_j^t are output token targets and predictions respectively, at turn t of output step j. In our case every token is treated equally to make the model as simple and as general as possible.

3.2.1 Attention mechanism

The signal from the user sentence can be amplified through the attention mechanism that enable focusing on the most relevant part of the utterance at each step. After each pass of the RNN over the user sentence the hidden vectors for each word are stored as:

$$\mathbf{h}^u = (\mathbf{h}_1^u, \mathbf{h}_2^u, ..., \mathbf{h}_L^u).$$

These are combined through a score function with the current hidden vector in the generation module $\mathbf{h}_{j-1}^m$:

$$a_{ij} = \text{score}(\mathbf{h}^u, \mathbf{h}_{j-1}^m) = \mathbf{v}^T \tanh(\mathbf{W}^T(\mathbf{h}_i^u \oplus \mathbf{h}_{j-1}^m) + \mathbf{b})$$

and passed through a softmax operator to produce the attention weights:

$$\alpha_{ij} = \frac{e^{a_{ij}}}{\sum_{i=1}^L e^{a_{ij}}}.$$

The current attentive hidden vector (the context vector) from the user sentence is created by weighting each hidden output with the attention weights:

$$\mathbf{h}_c^u = \sum_{i=1}^L \alpha_{ij} \mathbf{h}_i^u.$$

The context vector is concatenated with the previous generated embedded word

$$\mathbf{w}_{j-1}^{new} = \mathbf{h}_c^u \oplus \mathbf{w}_{j-1}$$

to output the prediction for the next word:

$$P(w_{j+1}|w_j, \mathbf{h}_{j-1}^m, \mathbf{h}^u) = \text{softmax}(\text{RNN}(\mathbf{w}_{j-1}^{new}, \mathbf{h}_{j-1}^m)).$$

3.3 Training set-up

All models were trained for 20 full passes over the training dataset. The best set of hyper-parameters was found by a grid search, evaluating on the held-out validation subset of the date. We varied between size of embedding vectors, hidden representation, $l2$-norm penalty and the size of the dictionary. The gradients are clipped by 5.0 throughout the whole training process. Since often times the evaluation of a dialogue system without a direct interaction with the real users can be misleading [Liu et al., 2016], three different automatic metrics are included to ensure the result is better interpreted. Among them, the first two metrics relate to the dialogue task completion - whether the system has provided an appropriate entity (Inform rate) and then answered all the requested attributes (Success rate); while fluency is measured via BLEU score [Papineni et al., 2002].

	Cam676		MultiWOZ	
	w/o attention	w/ attention	w/o attention	w/ attention
Inform (%)	99.17	99.58	71.29	71.33
Success (%)	75.08	73.75	60.29	60.96
BLEU	0.219	0.204	0.188	0.189

Table 1: Performance comparison of two different model architectures using a corpus-based evaluation.

3.4 Evaluation

Neural approaches to statistical dialogue development, especially in a task-oriented paradigm, are greatly hindered by the lack of large scale datasets. That is why, following the Wizard-of-Oz (WOZ) approach [Kelley, 1984, Wen et al., 2017b], we ran text-based multi-domain corpus data collection scheme through Amazon MTurk. The WOZ paradigm allowed us to obtain natural and semantically rich multi-topic dialogues spanning over multiple domains such as hotels, attractions, restaurants, booking trains or taxis. The dialogues cover from 1 up to 5 domains per dialogue greatly varying in length and complexity. We gathered around 10400 multi-domain dialogues of an average length and complexity substantially higher than in previous corpora. As expected, the model achieves almost perfect score on the Inform metric on the Cam676 dataset taking the advantage of an oracle belief state signal. However, even with the perfect dialogue state tracking of the user intent, the baseline models obtain almost 30% lower score on the Inform metric on the new corpus. The addition of the attention improves the score on the Success metric on the new dataset by less than 1%. Nevertheless, as expected, the best model on MultiWOZ is still falling behind by a large margin in comparison to the results on the Cam676 corpus taking into account both Inform and Success metrics. As most of dialogues span over at least two domains, the model has to be much more effective in order to execute a successful dialogue. Moreover, the BLEU score on the MultiWOZ is lower than the one reported on the Cam676 dataset. This is mainly caused by the much more diverse linguistic expressions observed in the MultiWOZ dataset.

4 Future work

Recently, [Wen et al., 2017a] employed a discrete latent variable to learn underlying dialogue intentions in the framework of neural variational inference. In a goal-oriented dialogue scenario, these latent intentions can be interpreted as actions guiding the generation of machine responses, which can be further refined autonomously by reinforcement learning. We plan to employ models presented in the previous section in order to create a model that is easier to learn and does not suffer from the *disconnection* phenomenon [Wen et al., 2017a]. The recent advances allows for creating only one loss that the model is optimized over contrary to previous approaches.

Moreover, the recently acquired dataset is annotated with system dialogue acts which enables pre-training in a supervised way [Su et al., 2017]. Initial analysis shows that over 55% of the annotated turns have multiple actions. These pose theoretical challenges requiring adapting standard RL framework to parallel action-choice [Harmer et al., 2018].

Finally, we are planning to incorporate the fine-tuning phase with dialogue-level RL loss. Previous methods used turn-level RL loss which results in an over-informative dialogue agent that does not learn multi-turn policy.

5 Conclusion

In this report, we have examined end-to-end dialogue modelling with neural architectures. Recently proposed methods were analyzed in the context of multi-domain dialogue systems. We created a multi-domain dialogue system with multi-domain belief state, database pointer and attention component over decoded space. The initial model was evaluated on the recently collected multi-domain dataset MultiWOZ. The initial framework is falling behind the performance of the model in the single-domain set-up. Multi-domain dialogues are longer often including two domains in one turn requiring better architectures that will allow for producing richer system outputs.

References

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *ICLR*.

[Bordes et al., 2017] Bordes, A., Boureau, Y.-L., and Weston, J. (2017). Learning end-to-end goal-oriented dialog. *Proceedings of ICLR*.

[Cho et al., 2014] Cho, K., Gulcehre, B. v. M. C., Bahdanau, D., Schwenk, F. B. H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Eric et al., 2017] Eric, M., Krishnan, L., Charette, F., and Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.

[Harmer et al., 2018] Harmer, J., Gisslén, L., Holst, H., Bergdahl, J., Olsson, T., Sjöö, K., and Nordin, M. (2018). Imitation learning with concurrent actions in 3d games. *CoRR*, abs/1803.05402.

[Kelley, 1984] Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

[Li et al., 2017] Li, X., Chen, Y.-N., Li, L., Gao, J., and Celikyilmaz, A. (2017). End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 733–743.

[Liu et al., 2016] Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Serban et al., 2015] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2015). Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*.

[Su et al., 2017] Su, P.-H., Budzianowski, P., Ultes, S., Gašić, M., and Young, S. J. (2017). Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the SIGDIAL 2017 Conference*.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Wen et al., 2017a] Wen, T.-H., Miao, Y., Blunsom, P., and Young, S. (2017a). Latent intention dialogue models. In *International Conference on Machine Learning*, pages 3732–3741.

[Wen et al., 2017b] Wen, T.-H., Vandyke, D., Mrksic, N., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. (2017b). A network-based end-to-end trainable task-oriented dialogue system. *EACL*.

[Young et al., 2013a] Young, S., Gašić, M., Thomson, B., and Williams, J. (2013a). POMDP-based Statistical Spoken Dialogue Systems: a Review. In *Proc of IEEE*, volume 99, pages 1–20.

[Young et al., 2013b] Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013b). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

[Young, 2002] Young, S. J. (2002). Talking to machines (statistically speaking).