Variational Cross-domain Natural Language Generation for Spoken Dialogue Systems

Bo-Hsiang Tseng, Florian Kreyssig, Paweł Budzianowski,
Iñigo Casanueva, Yen-Chen Wu, Stefan Ultes, Milica Gašić
Department of Engineering, University of Cambridge, Cambridge, UK
{bht26,flk24,pfb30,ic340,ycw30,su259,mg436}@cam.ac.uk

Abstract

Cross-domain natural language generation (NLG) is still a difficult task within spoken dialogue modelling. Given a semantic representation provided by the dialogue manager, the language generator should generate sentences that convey desired information. Traditional template-based generators can produce sentences with all necessary information, but these sentences are not sufficiently diverse. With RNN-based models, the diversity of the generated sentences can be high, however, in the process some information is lost. In this work, we improve an RNN-based generator by considering latent information at the sentence level during generation using the conditional variational autoencoder architecture. We demonstrate that our model outperforms the original RNN-based generator, while yielding highly diverse sentences. In addition, our model performs better when the training data is limited.

1 Introduction

Conventional spoken dialogue systems (SDS) require a substantial amount of hand-crafted rules to achieve good interaction with users. The large amount of required engineering limits the scalability of these systems to settings with new or multiple domains. Recently, statistical approaches have been studied that allow natural, efficient and more diverse interaction with users without depending on pre-defined rules (Young et al., 2013; Gašić et al., 2014; Henderson et al., 2014).

Natural language generation (NLG) is an essential component of an SDS. Given a semantic representation (SR) consisting of a dialogue act and a set of slot-value pairs, the generator should produce natural language containing the desired information.

Traditionally NLG was based on templates (Cheyer and Guzzoni, 2014), which produce grammatically-correct sentences that contain all desired information. However, the lack of variation of these sentences made these systems seem tedious and monotonic. *Trainable generators* (Langkilde and Knight, 1998; Stent et al., 2004) can generate several sentences for the same SR, but the dependence on pre-defined operations limits their potential. Corpus-based approaches (Oh and Rudnicky, 2000; Mairesse and Walker, 2011) learn to generate natural language directly from data without pre-defined rules. However, they usually require alignment between the sentence and the SR. Recently, Wen et al. (2015b) proposed an RNN-based approach, which outperformed previous methods on several metrics. However, the generated sentences often did not include all desired attributes.

The variational autoencoder (Kingma and Welling, 2013) enabled for the first time the generation of complicated, high-dimensional data such as images. The conditional variational autoencoder (CVAE) (Sohn et al., 2015), firstly proposed for image generation, has a similar structure to the VAE with an additional dependency on a condition. Recently, the CVAE has been applied to dialogue systems (Serban et al., 2017; Shen et al., 2017; Zhao et al., 2017) using the previous dialogue turns as the condition. However, their output was not required to contain specific information.

In this paper, we improve RNN-based generators by adapting the CVAE to the difficult task of cross-domain NLG. Due to the additional latent information encoded by the CVAE, our model outperformed the SCLSTM at conveying all information. Furthermore, our model reaches better results when the training data is limited.

2 Model Description

2.1 Variational Autoencoder

The VAE is a generative latent variable model. It uses a neural network (NN) to generate $\hat{x}$ from a latent variable z, which is sampled from the prior $p_\theta(z)$. The VAE is trained such that $\hat{x}$ is a sample of the distribution $p_D(x)$ from which the training data was collected. Generative latent variable models have the form $p_\theta(x) = \int_z p_\theta(x|z)p_\theta(z)dz$. In a VAE an NN, called the decoder, models $p_\theta(x|z)$ and would ideally be trained to maximize the expectation of the above integral $E\left[p_\theta(x)\right]$. Since this is intractable, the VAE uses another NN, called the encoder, to model $q_\phi(z|x)$ which should approximate the posterior $p_\theta(z|x)$. The NNs in the VAE are trained to maximise the variational lower bound (VLB) to $\log p_\theta(x)$, which is given by:

$$\begin{aligned} L_{VAE}(\theta, \phi; x) = -KL(q_\phi(z|x)||p_\theta(z)) \\ + E_{q_\phi(z|x)}[\log p_\theta(x|z)] \end{aligned} \quad (1)$$

The first term is the KL-divergence between the approximated posterior and the prior, which encourages similarity between the two distributions. The second term is the likelihood of the data given samples from the approximated posterior. The CVAE has a similar structure, but the prior is modelled by another NN, called the prior network. The prior network is conditioned on c. The new objective function can now be written as:

$$\begin{aligned} L_{CVAE}(\theta, \phi; x, c) = -KL(q_\phi(z|x, c)||p_\theta(z|c)) \\ + E_{q_\phi(z|x,c)}[\log p_\theta(x|z, c)] \end{aligned} \quad (2)$$

When generating data, the encoder is not used and z is sampled from $p_\theta(z|c)$.

2.2 Semantically Conditioned VAE

The structure of our model is depicted in Fig. 1, which, conditioned on an SR, generates the system's word-level response x. An SR consists of three components: the domain, a dialogue act and a set of slot-value pairs. *Slots* are attributes required to appear in x (e.g. a hotel's *area*). A *slot* can have a *value*. Then the two are called a *slot-value* pair (e.g. *area=north*). x is *delexicalised*, which means that slot values are replaced by corresponding slot tokens. The condition c of our model is the SR represented as two 1-hot vectors for the domain and the dialogue act as well as a binary vector for the slots.

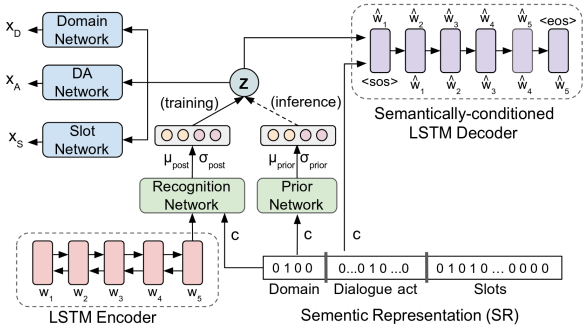


Figure 1: Semantically Conditioned Variational Autoencoder with a semantic representation (SR) as the condition. x is the system response with words $w_{1:N}$. x_D, x_A and x_S are labels for the domain, the dialogue act (DA) and the slots of x.

During training, x is first passed through a single layer bi-directional LSTM, the output of which is concatenated with c and passed to the recognition network. The recognition network parametrises a Gaussian distribution $\mathcal{N}(\mu_{post}, \sigma_{post})$ which is the posterior.The prior network only has c as its input and parametrises a Gaussian distribution $\mathcal{N}(\mu_{prior}, \sigma_{prior})$ which is the prior. Both networks are fully-connected (FC) NNs with one and two layers respectively. During training, z is sampled from the posterior. When the model is used for generation, z is sampled from the prior. The decoder is an SCLSTM (Wen et al., 2015b) using z as its initial hidden state and initial cell vector. The first input to the SCLSTM is a start-of-sentence (sos) token and the model generates words until it outputs an end-of-sentence (eos) token.

2.3 Optimization

When the decoder in the CVAE is powerful on its own, it tends to ignore the latent variable z since the encoder fails to encode enough information into z. Regularization methods can be introduced in order to push the encoder towards learning a good representation of the latent variable z. Since the KL-component of the VLB does not contribute towards learning a meaningful z, increasing the weight of it gradually from 0 to 1 during training helps to encode a better representation in z. This method is termed *KL-annealing* (Bowman et al., 2016). In addition, inspired by (Zhao et al., 2017), we introduce a regularization method using another NN which is trained to use z to recover the condition c. The NN is split into three separate FC NNs of one layer each, which independently

recover the *domain*, *dialogue-act* and *slots* components of c. The objective of our model can be written as:

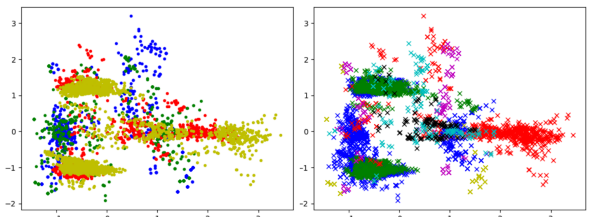
$$L_{SCVAE}(\theta, \phi; x, c) = L_{CVAE}(\theta, \phi; x, c) + E_{q_\phi(z|x,c)}[\log p(x_D|z) + \log p(x_A|z) + \log \prod_{i=1}^{|S|} p(x_{S_i}|z)] \quad (3)$$

where x_D is the domain label, x_A is the dialogue act label and x_{S_i} are the slot labels with $|S|$ slots in the SR. In the proposed model, the CVAE learns to encode information about both the sentence and the SR into z. Using z as its initial state, the decoder is better at generating sentences with desired attributes. In section 4.1 a visualization of the latent space demonstrates that a semantically meaningful representation for z was learned.

3 Dataset and Setup

The proposed model is used for an SDS that provides information about restaurants, hotels, televisions and laptops. It is trained on a dataset (Wen et al., 2016), which consists of sentences with corresponding semantic representations. Table 1 shows statistics about the corpus which was split into a training, validation and testing set according to a 3:1:1 split. The dataset contains 14 different system dialogue acts. The television and laptop domains are much more complex than other domains. There are around 7k and 13k different SRs possible for the TV and the laptop domain respectively. For the restaurant and hotel domains only 248 and 164 unique SRs are possible. This imbalance makes the NLG task more difficult.

The generators were implemented using the PyTorch Library (Paszke et al., 2017). The size of decoder SCLSTM and thus of the latent variable was set to 128. KL-annealing was used, with the weight of the KL-loss reaching 1 after 5k mini-batch updates. The slot error rate (ERR), used in (Oh and Rudnicky, 2000; Wen et al., 2015a), is the metric that measures the model's ability to convey the desired information. ERR is defined as: $(p + q)/N$, where N is the number of slots in the SR, p and q are the number of missing and redundant slots in the generated sentence. The BLEU-4 metric and perplexity (PPL) are also reported. The baseline SCLSTM is optimized, which has shown to outperform template-based methods and trainable generators (Wen et al., 2015b). NLG often uses the over-generation and reranking paradigm (Oh and Rudnicky, 2000). The SCVAE can generate multiple sentences by sampling multiple z, while the SCLSTM has to sample different words from the output distribution.In our experiments ten sentences are generated per SR. Table 4 in the appendix shows one SR in each domain with five illustrative sentences generated by our model.



(a) Domain label (b) Dialogue act label

Figure 2: 2D-projection of z for each data point in the test set, with two different colouring-schemes.

4 Experimental Results

4.1 Visualization of Latent Variable z

2D-projections of z for each data point in the test set are shown in Fig. 2, by using PCA for dimensionality reduction. In Fig. 2a, data points of the restaurant, hotel, TV and laptop domain are marked as blue, green, red and yellow respectively. As can be seen, data points from the laptop domain are contained within four distinct clusters. In addition, there is a large overlap of the TV and laptop domains, which is not surprising as they share all dialogue acts (DAs). Similarly, there is overlap of the restaurant and hotel domains. In Fig. 2b, the eight most frequent DAs are color-coded. `recommend`, depicted as green, has a similar distribution to the laptop domain in Fig. 2a, since `recommend` happens mostly in the laptop domain. This suggests that our model learns to map similar SRs into close regions within the latent space. Therefore, z contains meaningful information in regards to the domain, DAs and slots.

4.2 Empirical Comparison

4.2.1 Cross-domain Training

Table 2 shows the comparison between SCVAE and SCLSTM. Both are trained on the full cross-domain dataset, and tested on the four domains individually. The SCVAE outperforms the SCLSTM on all metrics. For the highly complex TV and laptop domains, the SCVAE leads to dramatic improvements in ERR. This shows that the addi-

Table 1: The statistics of the cross-domain dataset

	Restaurant	Hotel	Television	Laptop
# of examples	3114/1039/1039	3223/1075/1075	4221/1407/1407	7944/2649/2649
dialogue acts	reqmore, goodbye, select, confirm, request, inform, inform_only, inform_count, inform_no_match		compare, recommend, inform_all, suggest, inform_no_info, 9 acts as left	
shared slots	name, type, area, near, price, phone, address, postcode, pricerange		name, type, price, family, pricerange,	
specific slots	food, goodformeal, kids-allowed	hasinternet, acceptscards, dogs-allowed	screensizerange, ecorating, hdmiport, hasusbport, audio, accessories, color, screensize, resolution, powerconsumption	isforbusinesscomputing. warranty, battery, design, batteryrating, weightrange, utility, platform, driverange, dimension, memory, processor

Table 2: Comparison between SCVAE and SCLSTM. Both are trained with full dataset and tested on individual domains

Metrics	Method	Restaurant	Hotel	TV	Laptop	Overall
ERR(%)	SCLSTM	2.978	1.666	4.076	2.599	2.964
	SCVAE	**2.823**	**1.528**	**2.819**	**1.841**	**2.148**
BLEU	SCLSTM	0.529	0.642	0.475	0.439	0.476
	SCVAE	**0.540**	**0.652**	**0.478**	**0.442**	**0.478**
PPL	SCLSTM	2.654	3.229	3.365	3.941	3.556
	SCVAE	**2.649**	**3.159**	**3.337**	**3.919**	**3.528**

Table 3: Comparison between SCVAE and SCLSTM in K-shot learning

Metrics	Method	Restaurant	Hotel	TV	Laptop
ERR(%)	SCLSTM	13.039	**5.366**	24.497	27.587
	SCVAE	**10.329**	6.182	**20.590**	**20.864**
BLEU	SCLSTM	**0.462**	0.578	0.382	0.379
	SCVAE	0.458	**0.579**	**0.397**	**0.393**
PPL	SCLSTM	3.649	4.861	5.171	6.469
	SCVAE	**3.575**	**4.800**	**5.092**	**6.364**

tional sentence level conditioning through z helps to convey all desired attributes.

4.2.2 Limited Training Data

Fig. 3 shows BLEU and ERR results when the SCVAE and SCLSTM are trained on varying amounts of data. The SCVAE has a lower ERR than the SCLSTM across the varying amounts of training data. For very slow amounts of data the SCVAE outperforms the SCLSTM even more. In addition, our model consistently achieves better results on the BLEU metric.

4.2.3 K-Shot Learning

For the K-shot learning experiments, we trained the model using all training examples from three domains and only 300 examples from the target domain[1]. The target domain is the domain we test on. As seen from Table 3, the SCVAE outperforms the SCLSTM in all domains except hotel. This might be because the hotel domain is the simplest and the model does not need to rely on the knowledge from other domains. The SCVAE strongly outperforms the SCLSTM for the complex TV and laptop domains where the number of distinct SRs is large. This suggests that the SCVAE is better at transferring knowledge between domains.

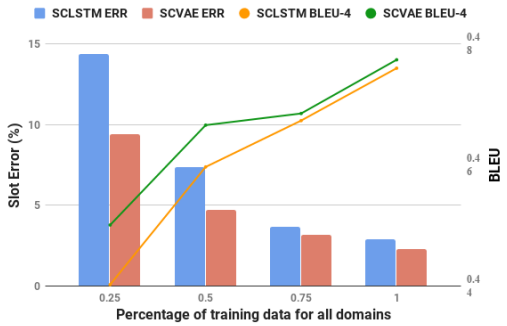


Figure 3: Comparison between SCVAE and SCLSTM with limited training data.

5 Conclusion

In this paper, we propose a semantically conditioned variational autoencoder (SCVAE) for natural language generation. The SCVAE encodes information about both the semantic representation and the sentence into a latent variable z. Due to a newly proposed regularization method, the latent variable z contains semantically meaningful information. Therefore, conditioning on z leads to a strong improvement in generating sentences with all desired attributes. In an extensive comparison the SCVAE outperforms the SCLSTM on a range of metrics when training on different sizes of data and for K-short learning. Especially, when testing the ability to convey all desired information within complex domains, the SCVAE shows significantly better results.

[1] 600 examples were used for laptop as target domain.

Acknowledgments

Bo-Hsiang Tseng is supported by Cambridge Trust and the Ministry of Education, Taiwan. This research was partly funded by the EPSRC grant EP/M018946/1 Open Domain Statistical Spoken Dialogue Systems. Florian Kreyssig is supported by the Studienstiftung des Deutschen Volkes. Paweł Budzianowski is supported by the EPSRC and Toshiba Research Europe Ltd.

References

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. *Generating Sentences from a Continuous Space.*

Adam Cheyer and Didier Guzzoni. 2014. Method and apparatus for building an intelligent automated assistant. US Patent 8,677,377.

M Gašić, Dongho Kim, Pirros Tsiakoulis, Catherine Breslin, Matthew Henderson, Martin Szummer, Blaise Thomson, and Steve Young. 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 360–365. IEEE.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 704–710. Association for Computational Linguistics.

François Mairesse and Marilyn A Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488.

Alice H Oh and Alexander I Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, pages 27–32. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *ACL*.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 79. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015a. Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015b. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664.

Table 4: Semantic representation (SR) with ground truth (GT) and sentences generated by SCVAE

Restaurant Domain	
SR	inform(name='la mediterranee';food='middle eastern';kidsallowed=no;pricerange=cheap)
GT	i have just the restaurant for you . it is called la mediterranee , it serves cheap middle eastern food and childs are not allowed
Gen1	la mediterranee serves middle eastern food in the cheap price range and does not allow childs
Gen2	la mediterranee is a cheap middle eastern restaurant that does not allow kids
Gen3	la mediterranee is cheaply priced restaurant serves middle eastern food and allow childs
Gen4	i would recommend la mediterranee . it is cheap middle eastern food , does not allow child
Gen5	la mediterranee does not allow kids , serves middle eastern food and it is cheap price
Hotel Domain	
SR	inform_count(type='hotel';count='2';near='marina cow hollow';pricerange='inexpensive')
GT	there are 2 other hotels near marina cow hollow that fit inexpensive your price range
Gen1	there are 2 inexpensive hotels near marina cow hollow
Gen2	there are 2 hotels near the marina cow hollow that are inexpensive
Gen3	there are 2 inexpensively priced hotel near marina cow hollow
Gen4	there are 2 inexpensive priced hotels near the marina cow hollow
Gen5	there are 2 hotels in the inexpensive price range near marina cow hollow
Television Domain	
SR	recommend(name=hymenaios 11;type=television;family=d1;hdmiport=1)
GT	the hymenaios 11 is a television with 1 hdmi port in the d1 product family
Gen1	the name of hymenaios 11 is a television in the d1 family with 1 hdmi port
Gen2	hymenaios 11 is a television from the d1 product family with 1 hdmi port
Gen3	the hymenaios 11 television is a member of the d1 product family and has 1 hdmi port
Gen4	the hymenaios 11 television is part of the d1 family , and has 1 hdmi port
Gen5	the hymenaios 11 is a nice television in the d1 family with 1 hdmi port
Laptop Domain	
SR	inform_no_match(type=laptop;isforbusinesscomputing=true;driverange=medium;weightrange=mid weight)
GT	no matches were found for a laptop search for business computing with a medium drive that falls in the mid weight range
Gen1	there is no laptop in the medium drive range that is mid weight range and are used for business use
Gen2	there are no laptops in the medium drive size range and is in the mid weight range . they are used for business computing
Gen3	sorry but there are no laptops that match for a medium drive range , mid weight weight range , and is for business computing
Gen4	there are no mid weight laptops with medium driver and are used for business computing
Gen5	unfortunately , we have no matches for your requirementss for a laptop in the mid weight range , that can be used for business computing , and is in the medium drive range