Unsupervised Discrete Sentence Representation Learning for Interpretable Neural Dialog Generation

Group: Gaussian's Confusion

December 7, 2019

Department of Computer Engineering
University of Virginia

Collaborators

Shimin Lei: Coding, PPT production, Result Analysis
Yining Liu: Coding, Data preprocessing, Result Analysis
Leizhen Shi: Coding, Data visualization
Hanwen Huang: Coding, Data collection

Code:

`https://github.com/ShiminLei/LA-Dialog-Generation-System`

Motivation and Background

Dialogue Act (DA)

Discourse structure is an important part for understanding dialogue, and plays a key role in dialog generation system. A useful way to describe discourse structure is identifying **Dialogue Act (DA)**, which represents the meaning of utterance at a level of illocutionary force [Stolcke et al., 2000].

Tag	**Example**
STATEMENT	I'm in the engineering department.
REJECT	Well, no.
OPINION	I think it's great.
AGREEMENT/ACCEPT	That's exactly it.
YES-NO-QUESTION	Do you have any special training?

Table: Dialogue Act Example

Conventional vs. Neural Dialog System

- Conventional dialog system: the action in a semantic frame usually contains hand-crafted dialog acts and slot values [Williams and Young, 2007]. But it's hard to design a fine-grained system manually.
- Neural dialog system: is a powerful frameworks without the need for hand-crafted meaning representations [Chung et al., 2014]. But it cannot provide interpretable system actions as in the conventional dialog systems.

Goal

Based on the importance of dialogue act interpretation and merits of neural dialog systems, the goal is to develop a neural network model which can discover interpretable meaning representations of utterances as a set of discrete latent variables (latent actions).

Related Work

Latent Variable Dialog Models

- The models proposed by [Vlad Serban et al., 2016] are based on Conditional Variational Autoencoders, where latent variables facilitate the generation of long outputs and encourage diverse responses.
- In the work discussed in [Zhao et al., 2017], dialog acts are further introduced to guide the learning of the Conditional Variational Autoencoders.
- For the recent research on task-oriented dialog system in [Wen et al., 2017], discrete latent variables have been used to represent intention .

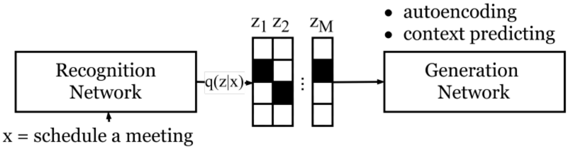
Sentence Representation Learning with Neural Networks

- Most work has been done for continuous distributed representations of sentences , e.g. the Skip Thought learns by predicting the previous and next sentences in [Kiros et al., 2015].
- Even passing gradients through discrete variables is very difficult, Gumbel-Softmax [Jang et al., 2016] make it possible to back-propagate by using continuous distribution sampling to approximate discrete distribution sampling.

Claim / Target Task and An Intuitive Figure Showing WHY Claim

Proposed Model

- Develop an unsupervised neural recognition model that can discover interpretable meaning latent actions from a large unlabelled corpus.



Proposed Model

Networks:

- Recognition network $\mathscr{R}$: $q_{\mathscr{R}}(\mathbf{z}|\mathbf{x})$

  Map an sentence to the latent variable $\mathbf{z}$

- Generation network $\mathscr{G}$

  Defines the learning signals that will be used to train the representation of $\mathbf{z}$.

The discovered meaning representations can be integrated with encoder decoder networks to achieve interpretable dialog generation.

Proposed Solution and Implementation

Learning Sentence Representations

Two methods:

- Learning Sentence Representations from Auto-Encoding
- Learning Sentence Representations from the Context

Learning Sentence Representations from Auto-Encoding

DI-VAE: Discrete infoVAE with BPR.

- Recognition network (RNN) last hidden state $h^{\mathcal{R}}_{|x|}$ represents $\mathbf{x}$.
- Define $\mathbf{z}$ to be a set of K-way categorical variables $\mathbf{z} = \{\mathbf{z_1} \ldots \mathbf{z_m} \ldots \mathbf{z_M}\}$
- For each $\mathbf{z_m}$, define its posterior distribution as $q_{\mathcal{R}}(\mathbf{z_m}|\mathbf{x})$. And we use the Gumbel-Softmax trick (a trick to solve the backpropagation problem for discrete variables) to sample from this distribution.
- Transform the latent samples $\mathbf{z_1} ... \mathbf{z_m}$ to $h^{\mathcal{G}}_0$, which is the initial state of Generation network (RNN).

Anti-Information Limitation of ELBO

VAEs often ignore the latent variable, especially when equipped with powerful decoders, which named as posterior collapse. To solve this problem, we decompose ELBO in a novel way to understand its behavior.

$$\begin{aligned}\mathscr{L}_{VAE} &= \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{q_{\mathscr{R}}(\mathbf{z}|\mathbf{x})}[\log_{p_{\mathscr{G}}}(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_{\mathscr{R}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - I(Z,X) - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))\end{aligned}$$

where $I(Z,X)$ is the mutual information, and $q(\mathbf{z}) = \mathbb{E}_{\mathbf{x}}[q_{\mathscr{R}}(\mathbf{z}|\mathbf{x})]$.

This shows that the KL term in ELBO is trying to reduce the mutual information between latent variables and the input data, which explains why posterior collapse happens.

VAE with Information Maximization and BPR

- Information Maximization: To correct the anti-information issue, we maximize both data likelihood lowerbound and the mutual information, so we optimize

$$\mathcal{L}_{VAE} + I(Z, X) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))$$

- Batch Prior Regularization (BPR): To minimize the $\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))$.

  Let $\mathbf{x}_n$ be a sample from a batch of N data points, we have

$$q(\mathbf{z}) \approx \frac{1}{N}\sum_{n=1}^{N} q(\mathbf{z}|\mathbf{x}_n) = q'(\mathbf{z})$$

  We can approximate $\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))$ by

$$\mathrm{KL}(q'(\mathbf{z})||p(\mathbf{z})) = \sum_{k=1}^{K} q'(\mathbf{z}=k)\log\frac{q'(\mathbf{z}=k)}{p(\mathbf{z}=k)}$$

  This equation is referred as BPR.

Learning Sentence Representations from the Context

DI-VST: DI-VAE to Discrete Information Variational Skip Thought.

- Skip thought (ST): The meaning of language can be inferred from the adjacent context.
- Use the same recognition network from DI-VAE to output $\mathbf{z}$'s posterior distribution $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$.
- Given the samples from $q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})$, two RNN generators are used to predict the previous sentence $\mathbf{x}_p$ and the next sentences $\mathbf{x}_n$.
- Objective to maximize

$$\mathcal{L}_{DI-VST} = \mathbb{E}_{q_{\mathcal{R}}(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log(p^n_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z})p^p_{\mathcal{G}}(\mathbf{x}_p|\mathbf{z}))] - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))$$

Data Summary

Datasets

The proposed methods are evaluated on five datasets.

- Penn Treebank (PTB)
- Stanford Multi-Domain Dialog (SMD)
- Daily Dialog (DD)
- Switchboard (SW)
- Multimodal EmotionLines Dataset (MELD)

Reproduction Experimental Results and Analysis

Comparing Discrete Sentence Representation Models

Part 1: Evaluate proposed model performance

For comparison, we use several baselines model.

Unregularized models:

- DAE: Remove the KL($q|p$) term from DI-VAE.
- DST: Remove the KL($q|p$) term from DI-VST.

ELBO models: (KL-annealing and bag-of-word loss used)

- DVAE (posterior collapse): The basic discrete sentence VAE that optimizes the ELBO with regularization term KL($q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})$).
- DVST (posterior collapse): The basic discrete sentence variational skip thought that optimizes the ELBO with regularization term KL($q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})$).

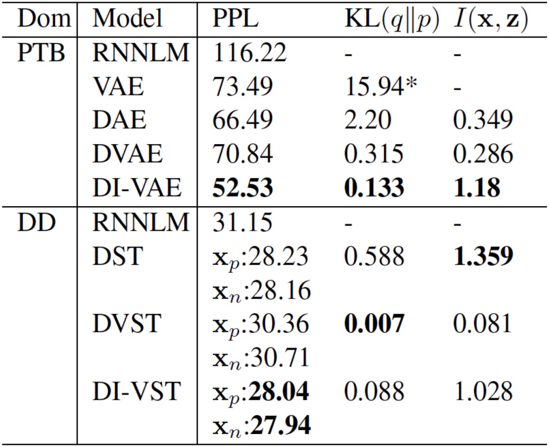
Comparing Discrete Sentence Representation Models

Other models:

- VAE: VAE with continuous latent variables (results by Zhao et al., 2017).
- RNNLM: Standard GRU-RNN language model (results by Zaremba et al.,2014).

Comparing Discrete Sentence Representation Models

The comparing results (with the discrete latent space for all models are M=20 and K=10 and Mini-batch size is 30):



Dom	Model	PPL	$\mathrm{KL}(q\|p)$	$I(\mathbf{x}, \mathbf{z})$
PTB	RNNLM	116.22	-	-
	VAE	73.49	15.94*	-
	DAE	66.49	2.20	0.349
	DVAE	70.84	0.315	0.286
	DI-VAE	**52.53**	**0.133**	**1.18**
DD	RNNLM	31.15	-	-
	DST	$\mathbf{x}_p$:28.23 $\mathbf{x}_n$:28.16	0.588	**1.359**
	DVST	$\mathbf{x}_p$:30.36 $\mathbf{x}_n$:30.71	**0.007**	0.081
	DI-VST	$\mathbf{x}_p$:**28.04** $\mathbf{x}_n$:**27.94**	0.088	1.028

Reproduction Result

Dom	Model	PPL	KL$(q\|\|p)$	$I(x,z)$
PTB	DAE	63.443	1.671	0.514
	DVAE	73.744	0.249	0.025
	DI-VAE	**52.751**	**0.130**	**1.207**
MELD	DAE	55.884	2.047	0.237
	DVAE	92.893	0.060	0.055
	DI-VAE	**44.800**	**0.054**	**1.005**
DD	DST	x_p:28.967/x_n:29.659	2.303	0.000
	DVST	x_p:87.964/x_n:90.818	**0.023**	0.004
	DI-VST	x_p**:28.073/**x_n**:28.085**	0.084	1.015
MELD	DST	x_p:68.237/x_n:69.367	2.303	0.000
	DVST	x_p:88.166/x_n:88.148	0.032	0.002
	DI-VST	x_p**:67.324/**x_n**:68.778**	0.007	0.099

Comparing Discrete Sentence Representation Models

Analysis:

- All models achieve better perplexity than an RNNLM, which shows they manage to learn meaningful $q(\mathbf{z}|\mathbf{x})$.
- DI-VAE achieves the best results in all metrics compared others.
- DI-VAE vs. DAE:
  1. DAE learns quickly but prone to overfitting.
  2. For DAE, since there is no regularization term in the latent space, $q(\mathbf{z})$ is very different from the $p(\mathbf{z})$, which prohibits us from generating sentences from the latent space.

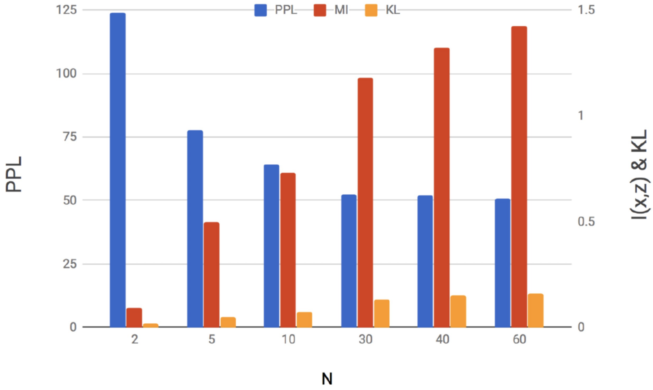
Comparing Discrete Sentence Representation Models

- DI-VST vs. DVST and DST:
  1. DI-VST is able to achieve the lowest PPL.
- These results confirm the effectiveness of the proposed BPR in terms of regularizing $q(\mathbf{z})$ while learning meaningful posterior $q(\mathbf{z}|\mathbf{x})$.

Comparing Discrete Sentence Representation Models

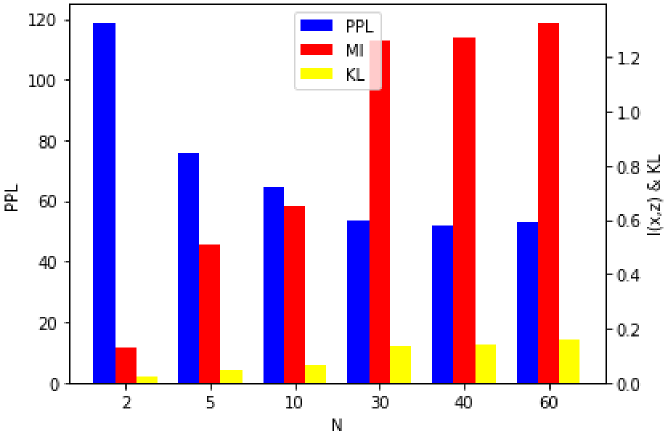
Part 2: Understand BPR's sensitivity

In order to understand BPR's sensitivity to batch size N, we varied the batch size from 2 to 60 (If N=1, DI-VAE is equivalent to DVAE).



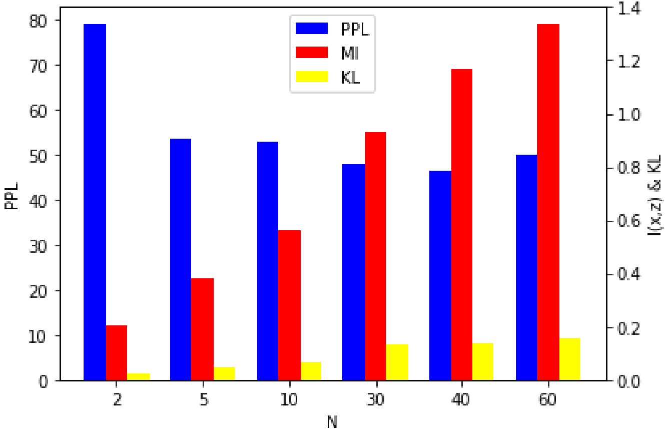
Reproduction Result

For PTB dataset:



Reproduction Result

For MELD dataset:



Comparing Discrete Sentence Representation Models

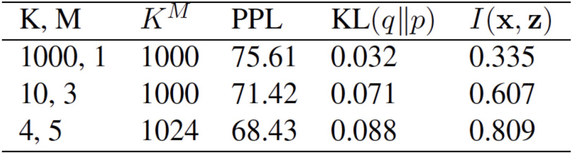
Analysis:

- As N increases, perplexity, $I(\mathbf{x}, \mathbf{z})$ monotonically improves, while $\text{KL}(q||p)$ only increases from 0 to approximate 0.16.
- After $N > 30$, the performance plateaus. Therefore, using mini-batch is an efficient trade-off between $q(\mathbf{z})$ estimation and computation speed.

Comparing Discrete Sentence Representation Models

Part 3: Relation between representation learning and the dimension of the latent space

We set a fixed budget by restricting the maximum number of modes to be about 1000, i.e. $K^M \approx 1000$.



K, M	K^M	PPL	$\mathrm{KL}(q\|p)$	$I(\mathbf{x}, \mathbf{z})$
1000, 1	1000	75.61	0.032	0.335
10, 3	1000	71.42	0.071	0.607
4, 5	1024	68.43	0.088	0.809

Reproduction Result

For PTB dataset:

K,M	K^M	PPL	$\mathrm{KL}(q\|\|p)$	$I(x,z)$
1000,1	1000	76.240	0.028	0.254
10,3	1000	72.815	0.054	0.539
4,5	1024	67.537	0.079	0.757

For MELD dataset:

K,M	K^M	PPL	$\mathrm{KL}(q\|\|p)$	$I(x,z)$
1000,1	1000	67.567	0.000	0.004
10,3	1000	65.051	0.017	0.440
4,5	1024	61.214	0.013	0.418

Analysis: Models with multiple small latent variables perform significantly better than those with large and few latent variables.

Interpreting Latent Actions

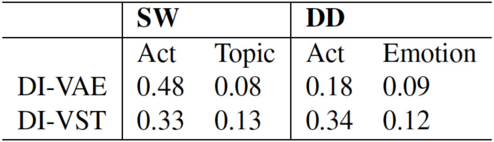
The question is to interpret the meaning of the learned latent action symbols. The latent action of an utterance of $\mathbf{x}_n$ is obtained from a greedy mapping:

$$a_n = \arg\max_k q_{\mathcal{R}}(\mathbf{z} = k|\mathbf{x}_n)$$

We set M=3 and K=5, so there are at most 125 different latent actions, and each $\mathbf{x}_n$ can be represented by $a_1 \rightarrow a_2 \rightarrow a_3$, e.g. "How are you ?" → 1-4-2.

Interpreting Latent Actions

For manually clustered data: We utilize the homogeneity metric that measures if each latent action contains only members of a single class.



	SW		DD	
	Act	Topic	Act	Emotion
DI-VAE	0.48	0.08	0.18	0.09
DI-VST	0.33	0.13	0.34	0.12

Summary: For acts, DI-VST performs better on DD and worse on SW than DI-VAE. One reason is that the dialog acts in SW are more fine-grained (42 acts) than the ones in DD (5 acts) so that distinguishing utterances based on words in x is more important than the information in the neighbouring utterances.

Reproduction Result

For DailyDialog Dataset with K=10, M=10:

	DD	
	Act	Emotion
DI-VAE	0.15972	0.10352
DI-VST	0.13797	0.07356

Analysis: The homogeneity of Act is larger than that of Emotion, which indicates that the model can capture the attribute of latent action better.

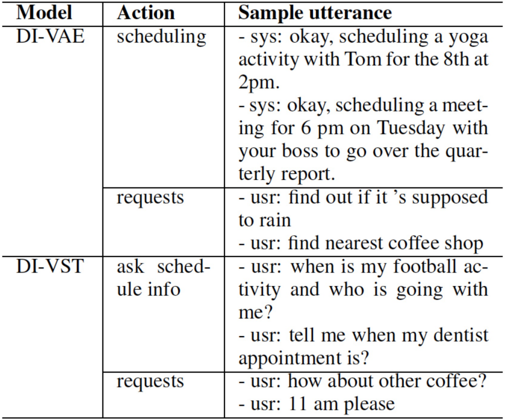
Interpreting Latent Actions

Other Analysis:

- Since DI-VAE is trained to reconstruct its input and DI-VST is trained to model the context, they group utterances in different ways.
- For example, DI-VST would group “Can I get a restaurant”, “I am looking for a restaurant” into one action where DI-VAE may denote two actions for them.

Interpreting Latent Actions

- An example latent actions discovered in SMD using the methods.



Model	Action	Sample utterance
DI-VAE	scheduling	- sys: okay, scheduling a yoga activity with Tom for the 8th at 2pm. - sys: okay, scheduling a meeting for 6 pm on Tuesday with your boss to go over the quarterly report.
	requests	- usr: find out if it 's supposed to rain - usr: find nearest coffee shop
DI-VST	ask schedule info	- usr: when is my football activity and who is going with me? - usr: tell me when my dentist appointment is?
	requests	- usr: how about other coffee? - usr: 11 am please

Conclusion and Future Work

Conclusion

- This paper presents a novel unsupervised framework that enables the discovery of discrete latent actions and interpretable dialog response generation.
- The main contributions reside in the two sentence representation models DI-VAE and DIVST, and their integration with the encoder decoder models.
- Experiments show the proposed methods outperform strong baselines in learning discrete latent variables and showcase the effectiveness of interpretable dialog response generation.

Future Work

- The findings suggest promising future research directions, including learning better context-based latent actions and using reinforcement learning to adapt policy networks.
- This work is an important step forward towards creating generative dialog models that can not only generalize to large unlabelled datasets in complex domains but also be explainable to human users.

Reference

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014).
Empirical evaluation of gated recurrent neural networks on sequence modeling.
CoRR, abs/1412.3555.

Jang, E., Gu, S., and Poole, B. (2016).
Categorical Reparameterization with Gumbel-Softmax.
arXiv e-prints, page arXiv:1611.01144.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015).
Skip-thought vectors.
CoRR, abs/1506.06726.

Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Van Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., and Meteer, M. (2000).
Dialogue act modeling for automatic tagging and recognition of conversational speech.
Comput. Linguist., 26(3):339–373.

Vlad Serban, I., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. (2016).
A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues.
arXiv e-prints, page arXiv:1605.06069.

Wen, T.-H., Miao, Y., Blunsom, P., and Young, S. (2017).
Latent Intention Dialogue Models.
arXiv e-prints, page arXiv:1705.10229.

Williams, J. D. and Young, S. (2007).
Partially observable markov decision processes for spoken dialog systems.
Computer Speech Language, 21(2):393 – 422.

Zhao, T., Zhao, R., and Eskénazi, M. (2017).
Learning discourse-level diversity for neural dialog models using conditional variational autoencoders.
CoRR, abs/1703.10960.

Thank you!