Generating Controllable Long-dialogue with Coherence

Zhecheng Sheng, Chen Jiang and **Tianhao Zhang**
Team Sprite
{sheng136,jian0649,zhan7594}@umn.edu

Abstract

Developing controllable long text generation model has been a challenging problem given its relatively complex syntactic and semantic structures comparing to short form text. In this project, we aim to study approaches that can manage the coherence and cohesion from generating long text and apply it to open domain dialogue system. The targeted outcome will be a model feasible to generate long coherent dialogues with dynamic information flows and imposed keywords related content. Our work fundamentally rely on the stochastic modeling paper (Wang et al., 2022). Within the scope of course project, we firstly examine the characteristics of the latent variables and produced some explanations for our observations. We also did extensive experiments checking how the latent trajectories relates to the coherence of generated text and developed or applied several evaluation metrics for coherence to test it.

1 Motivation

Modern open dialogue systems can benefit from the large amount of free corpus to train end-to-end neural models(Budzianowski and Vulić, 2019), however they also suffer from generating repetitive, meaningless, sometimes irrelevant text due to their over-simplified training objectives. On the other hand, large language models have been proven to generate short text effectively, but they are often incoherent when it comes to longer text due to their inability to plan-ahead and represent long-range text dynamics. One recent work (Wang et al., 2022) proposed a representation-learning based method, named *Time Control*, to learn a latent space with known, target-anchored fixed dynamics. The authors made two explicit assumptions in the paper. They begin by assuming the wandering text produced without a target can be modeled by Brownian motion, which enforces the embeddings of neighboring text to be similar to each other and distant text to be dissimilar. The second assumption is that goal-oriented generation can be achieved by fixing start and end point. Thus the latent trajectories can be guided by simple, closed-form dynamics. The experiment results show promising tail end quality in forced long text generation. We are hoping to extend this stochastic language modeling to be more specific for dialogue generation and append other information such dynamic flow and extra knowledge to get more knob-tweaking ability during the generation procedure.

2 Literature Survey

2.1 Coherent long text generation

There has been previous efforts that try to address the uncontrolled aspects in long text generation. One category is called prompt control where a context is prepended to a model input. For example, CTRL(Keskar et al., 2019) is a large conditional transformer language model that combine control codes and prompts to exhibit a style of text. However, prompt-based methods suffer from the inability to iteratively improve the existing generated text. Various works have used syntactic parse tree with the transformer structure to gain discourse control and enhance interpretability. RSTGen (Adewoyin et al., 2022) applies RST theory to form a binary tree structure of discourse unit and shows superiority in long-form text generation task such as argument generation and story generation. Planning based approaches has been used to generate globally coherent text(Kiddon et al., 2016), but they often requires domain-specific knowledge to define dynamics. Learning latent representation for controlling long text generation is also an approach that many works have been studied, but they either cannot capture evolving dynamics over the whole document (Nie et al., 2019) or difficult to apply learned local dynamics to generate accurate goal-conditioned trajectory(Oord et al., 2018).

2.2 Knowledge enriched text generation

The knowledge can be divided into two classes based on the knowledge source. One is external knowledge from outside such knowledge graph or knowledge base. The other is internal knowledge embedded in input text. Internal knowledge sources can be further divided into topic, keyword, linguistic features and internal graph structure (Yu et al., 2022). In this project, we are going to focus on using topic and keyword to enhance natural language generation.

In dialogue system, a vanilla Seq2Seq model tends to output trivial replies with little information and can often lead the conversation to an end easily. Generative topic models were proposed to generate on-topic response (Xing et al., 2017; Zhang et al., 2016; Liu et al., 2019). They first generate topic from a pretrained generative model, for example Latent Dirichlet Allocation (LDA), and then feed the topic representation to neural generation models. However, the topic generation process is separated from neural network training. It is difficult for them to capture the complicated relationship between input and output text. Instead, neural topic models can be trained efficiently be backpropagation and provide better coherence (Cao et al., 2015; Miao et al., 2017).

Keyword enhanced methods can be categorized in to keyword assignment methods and keyword extraction methods. While keyword extraction selects the existing words in input text as keyword, keyword assignment uses a classifier to assign one word in predefined vocabulary to be keyword (Siddiqi and Sharan, 2015). Keyword extraction was shown to be suitable for keeping important information in text such as summarization and paraphrase (Yu et al., 2022). Keyword assignment is more popular in dialogue system. It has been used to incorporate emotion or persona into generated response and get superior performance (Li and Sun, 2018; Song et al., 2019a,b; Xu et al., 2020).

2.3 Utilize dialogue history

Currently, the dialogue history model for open-domain dialogue can be generally classified to the following two types: 1) Flat pattern model: the predicted response is generated by concatenate the dialogue history as the model input (Zhang et al., 2020). This method is commonly used in the large-scale pre-training. 2) Hierarchical model: to generated the predicted response, one needs to first encode each utterance separately and then the predicted response is generated by feeding all these encoded utterances into an utterance-level encoder (Gu et al., 2021).

There are some drawbacks of the above two models. For the flat pattern model, it is likely to ignore the conversational dynamics across the utterances in the dialogue history (Sankar et al., 2019). For the hierarchical model, in the utterance encoder, it fails to make use of the history information, and instead, encode each utterance individually. To overcome this two deficiencies, inspired by human cognitive process, (Li et al., 2021b) put forward the DialoFlow to model the dynamic information flow in the dialogue history by addressing the semantic influence brought about by each utterance.

Besides, considering history information is often be long and noisy, it may be a waste of time and energy to work on all of the information and such redundant information may lead to system to a sub-optimal solution. Instead, a refined dialogue history information set may enhance the model performance, among the method to refine the history information, one can use the keywords(Li et al., 2021a), tokens (Zhong et al., 2022) and other contextual knowledge.

2.4 Evaluation metrics

Based on our literature review, there exists several automatic metrics for measuring local coherence of the text. However, since most of these metrics were developed for machine translations, it requires a reference text to be compared with. In our generation process from latent space, the learned embedding does not necessarily produce matched text with the input ground truth in any details throughout the dialogue, so lots of existing automatic metrics are not suitable for our settings and is also our major challenge. In our experiments, we will use two model-based approaches to evaluate our test cases (chatGPT as an extra in the appendix). We will also provide human evaluations and discuss its correlation with the chosen automatic evaluation metrics.

2.4.1 Automatic Evaluation

To automatically evaluate the text coherence problem is a long standing problem. In our task, we don't have reference to do the evaluation, so the currently popular score, such as BLEU(Papineni et al., 2002), METEOR(Banerjee and Lavie, 2005), NIST or BERTScore(Zhang* et al., 2020) cannot be used in our task. Besides, since our dialogue generation model has a preset topic, therefore, we can use some topic-related automatic evaluation metric. Due to the time limitation, here we only show on automatic evaluation metric for text coherence. Here, we use the automatic text coherence evaluation put forward in (Röder et al., 2015). In this paper, they put forward a framework that make use of the existing conformation measures which can be used to evaluate the text coherence. In our paper, we select three difference conformation measures: UCI coherence, U-Mass coherence and Normalized pointwise mutual information.

2.5 Datasets

We will consider the following datasets in our experiments.

Datasets	Training size
Wikisection	1420 articles
Wikihow	1566 articles
Taskmaster-2	2000 conversations
TicketTalk	2000 conversations

3 Proposed Ideas

3.1 Problem definition

Here we want to address the research questions from three aspects:

RQ1: Can Time Control robustly generate long and coherent dialogues?

RQ2: Will incorporate dialogue history help improve the generation performance?

RQ3: Can we further impose keyword or topic to guide the content generation?

3.2 Topic and keyword model

We can use neural topic model to generate latent topic and feed it into the decoder along with latent text to get on-topic responses. Keyword assignment methods can also be adopted to incorporate personalized information such as emotion and persona to responses.

3.3 Enhanced model performance by dialogue history

In (Wang et al., 2022)'s Time Control model, when they predict the trajectory $\{z_t\}_{t=0}^{T}$ in the latent space, the only history information they use is the starting point z_0 and the ending point z_T[1]. Due to this problem, their dialogue model easily forgets what the people were just talking about, for example, [U]: "I want five tickets" [A]: "Okay, I found ..." [U]:"..." [A]: "How many tickers would you like to purchase?" (Wang et al., 2022). Another problem in this dialogue model which is probably caused by this issue is the lack of a coherent theme, for example, the user started the dialogue by saying he wants to buy a ticket to movie A, but as the dialogue continues, they seems forget their original idea and talked about the movie B.

We attribute these problems to insufficient use of historical information. To enhance the model performance, we will try to add the history information to the response generation part. However, considering the intrinsic problem cause by the Brownian bridge assumption, that the predicted latent variables $\{z_t\}_{t=1}^{T-1}$ only depends on z_0 and z_T, we can only add an intermediate model to process all the predicted latent variables before inputting them to the decoder, instead of working on the generation mechanism of the latent variables. However, this can be regarded as a hierarchical model, which faces the same problem we discussed in the literature part. To overcome this problem, we may adopt idea from the DialoFlow model (Li et al., 2021b), that is to construct another model to capture the overall history information and the dynamic of the dialogue. Furthermore, such idea will be refined as the project progresses, for example, we may refine the history information set with the keywords.

Time Control : [USER] I'm looking for movie tickets please. [ASSISTANT] OK, where would you like to see the movie? [USER] I'd like to see it at Creek's End, Oregon please. [...] [ASSISTANT] Is it OK to go ahead and purchase these tickets? [USER] Yeah, that would be great. [ASSISTANT] OK. I understand that you are interested in tickets for Emma at AMC Mercado 24 tonight at 7:10pm. Is that OK? [USER] Yes, please do that. [ASSISTANT] OK

GPT2: [USER] Hi! Tonight, I'd like to go to the movies. [ASSISTANT] Okay. What theater would you like to go to? [USER] Center City. [...] [ASSISTANT] That will be all for now. Thank you for all your help. N/A [USER] Bye Bye. [ASSISTANT] N/A [ASSISTANT] N/A N/A N/A N/A N/A N/A N/A N/A [USER] N/A [ASSISTANT] N/A N/A N/A [USER] N/A [ASSISTANT] N/A N/A N/A [USER] N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A [USER] N/A N/A N/A N/A [...]

Figure 1: Paper Example

4 Proposed Analysis

We dive into the latent space learned through a Brownian bridge and explore how the change in shape of the latent space can affect the coherence of generated text. Though our anticipated outcome for the project is to develop a text generation system suitable for open-domain chit-chat talk, the example1 shown on the original paper is dedicated to a static latent space trained on certain dataset (i.e. TicketTalk), so our initial exploration focus on task-oriented long text generation and assess

[1]If the real end-point does not exist, this point will be generated randomly.

its coherence. But eventually we aim to employ stochastic process language model approach into traditional sequence to sequence model in interactive dialogue mode which we think is a greater potential to be used and has greater flexibility in real life. The high level idea for language model via stochastic process is to map raw text into a latent space via an encoder, which utilizes Brownian bridge as part of the loss function. To examine the relations between long text coherence and the shape of the latent space, we plan to address from the following aspects. In section 4.1, we describe how the latent space is learned in detail. Additionally, in section 4.2, we will discuss how to set up experiments to quantify the latent variable through visualization, and whether sampling different trajectories from the latent space impact the coherence of the generated text. We present our results in section 4.3.

4.1 Latent space construction

The intuition of using stochastic process to model the text generation is to learn the smooth temporal dynamics of a latent space. The original paper uses a *GPT2* encoder with a multi-layer perceptron on top of it. According to the paper(Wang et al., 2022), the encoder architecture is trained via a non-linear function to map sentences to a Brownian bridge latent space, $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$. That way we can map high dimensional sequence into arbitrarily lower dimension. Given a arbitrary starting point z_0 and end point z_T, the Brownian bridge is defined as:

$$p(z_t|z_0, z_T) = \mathcal{N}\left((1 - \frac{t}{T})z_0 + \frac{t}{T}z_T, \frac{t(T-t)}{T}\right)$$

The density can be viewed as a noisy linear interpolation between two anchored points, where the variance grows towards the middle region and shrinks towards the end.

In terms of the training steps, let (x_1, x_2, x_3) be a triplet observations from the training set, the objective is to ensure $f_\theta(x_1), f_\theta(x_2), f_\theta(x_3)$ follow the transition probability defined in Brownian bridge. Positive triples (x_0, x_t, x_T) can be randomly sampled from a whole sequence of data points $X = \{x_1, x_2, .., x_n\}$ as long as $0 < t < T$ on the original order and the loss function to optimize is:

$$\mathcal{L}_N = \mathbb{E}_X[-log \frac{exp(d(x_0, x_t, x_T; f_\theta))}{\sum_{(x_0, x_{t'}, x_T)\in\mathcal{B}} exp(d(x_0, x_{t'}, x_T; f_\theta))}]$$

$$d(x_0, x_t, x_T; f_\theta) = -\frac{1}{2\sigma^2}||f_\theta(x_t) - (1 - \frac{t}{T})f_\theta(x_0) - \frac{t}{T}f_\theta(x_T)||$$

It is worth to note in the denominator of the loss function, it is summing over all negative middle contrast $x_{t'}$, which can be seen as the mid-point sample drawn from another alternative sequence.

Once the encoder is optimized, a decode is supposed to be trained for learning generation. Basically the decoder is a fine tuned *GPT2* and generate text conditioned all past context and the learned latent plan. For now, we are only going to study the attributes of the latent plan created by the encoder, thus skip the detailed decoder and generation sections .

4.2 Experiment setup

To study the latent plan and its correlation with the coherence of the text, we first designed 3 experiments as the exploratory analysis:

- Quantify the distribution of latent variable through visualizing different dimensions.
- Explore the relations between latent space distribution and text coherence by altering the distribution of the latent space.
- Explore the effect of extreme cases by pining the trajectory and examine the text coherence.

We trained encoders from scratch and fine tuned decoders using four different dataset: Wikisection, Wikihow, TicketTalk and TicketMaster2. We used 16 latent dimensions for training Wikihow, TicketTalk and TicketMaster2 dataset while for Wikisection we used 32 latent dimensions during training. We kept other hyperparameters during encoder training consistent across four dataset and they are set as the followings:*{model_params: {num_layers: 2, hidden_size: 128, eps: 1*e^{-6}*}, optimiz_params:{batch_size:32, decay_step: 5*e^4*, decay_factor: 0.1, learning_rate: 0.0001, moving_average_decay: 0.9999, momentum: 0.9}, expriment_params:{num_epochs:100}}*. And all the training was ran using a single GPU *NVIDIA GeForce RTX 3090 Ti*.

For evaluations, we generated 5 documents for each experiment settings from 2 different conversation dataset. Specifically, those experiment settings are: 1. True embeddings of the sample 2. Brownian bridge embeddings sampled from a learned Brownian Bridge process with fixed starting and ending point. 3. Brownian bridge embeddings with foced trajectories (enlarge the variance by factor of 2 and 3) 4. Random embeddings sampled from Gaussian distributions with the same dimension.

4.3 Results

4.3.1 Visualization of the latents

To show whether the latents follow Brownian bridge process, we plot the trajectories of latents generated by trained Time Control encoders. Since the Time Control model assumes that all dimensions of the latent variable is independent of each other, we can draw each dimension separately and then analyze their distributions. Figure 2 shows the plots of each dimension of latents generated from one text in TicketTalk training set. Surprisingly all dimensions share similar values. Every dimension is either close to dimension 0, or close to the negative of dimension 0. Also we observe that, there is a big jump near the end of the text, which is a common phenomenon in all the plots from different dataset or data points. If we remove the last several latents from the plots, the remaining trajectory looks like coming from a Brownian bridge process.

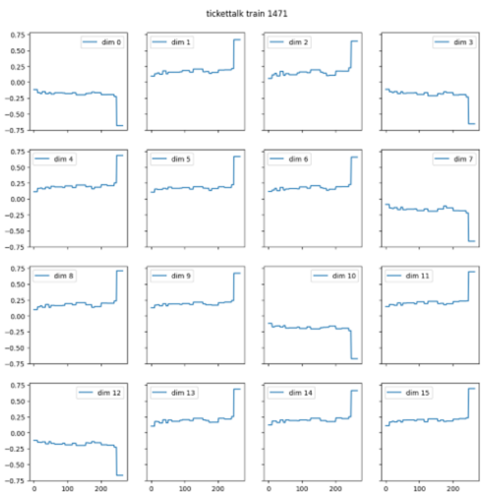


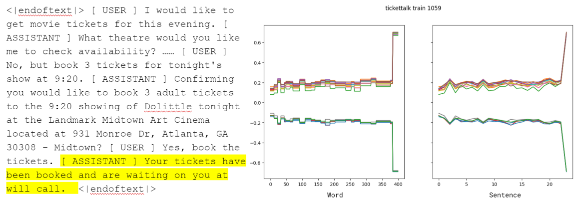
Figure 2: Plots of Each Dimension of Latents

To analyze why there are big jumps in latent trajectories, we present several latent trajectories and corresponding raw texts. Three common patterns of big jumps is shown in Figure 3. They are all near the end of trajectories and usually come from the sentences about confirmation of booked tickets and concluding the conversation.

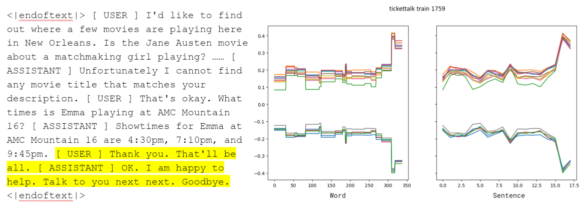
As we can see in Figure 4, the latent trajectories generated from other datasets, including Wikisection and TicketMaster2, also have similar behaviors as above.

4.3.2 Extreme trajectory

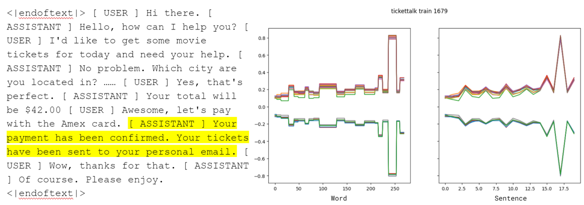
In this subsection, we test the model behavior by manually selecting the latent trajectory to be some



(a) Big Jump in Last Sentence

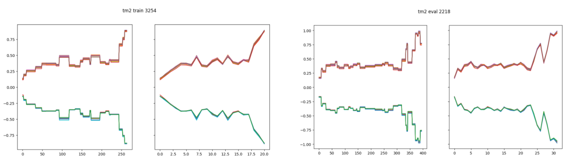


(b) Big Jump in Last Two Sentence

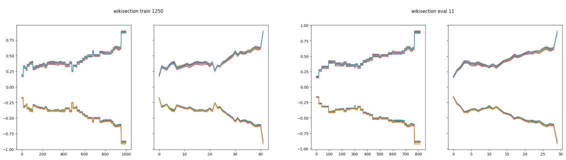


(c) Big Jump Near the End

Figure 3: Raw Texts with Word-level and Sentence-level Latent Trajectories



(a) TicketMaster2



(b) Wikisection

Figure 4: Latent Trajectories from Other Datasets

extreme trajectory (boundary of sampling region). Here we show one example in Figure 5 where the trajectory is manually selected to be boundary of the initial sampling region, and the rest tests can be found in the appendix 7.

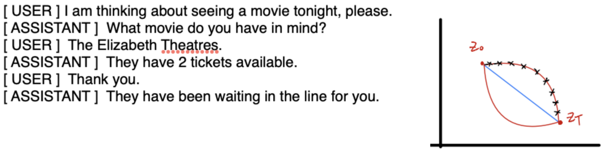


Figure 5: The region enclosed by the red curve is the sampling region of latent trajectory. The trajectory consists of black crosses is the latent sequence corresponds to the text.

In fact, the boundary are possibly the worst cases when sampling in the corresponding sampling region. By analyzing these worst cases, even though these extreme cases are less likely to be generated in application, we can still get a very shallow intuition on the model stability and are able to find a faint relation between latent space and text coherence. To further justify this intuition, we do the following tests in section 4.3.3.

4.3.3 Extended sampling region

In this subsection, we evaluate the relation between size of sampling region and text coherence. Here we show one example in Figure 6, and the rest can be found in the appendix 7. In this test, the sampling region is expanded to twice the original region, and the latent sequences are sampling in this region. As we can see, the text coherence is worse than the example shown in Figure 8 which is corresponding to the initial sampling region.

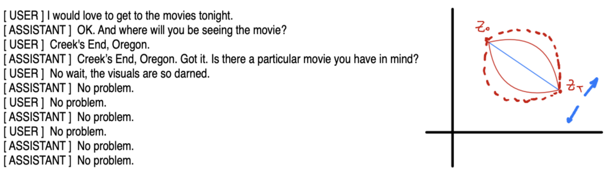


Figure 6: The region enclosed by the red full curve is the initial sampling region. The region enclosed by the red dash curve is the extended sampling region of this test.

By simply comparing these results, we can find a preliminary result that the initial region without any extension generates the most coherent text, and if the size of sampling region increases, the text coherence will decrease. However, even though we have already generated many other cases that also show this property, we still need a non-human evaluation so that we can test on a large amount of cases to eliminate the impact of randomness.

4.4 Evaluation results

4.4.1 Semantic Coherence

Inspired by the method of using clustering algorithms on pre-trained word embeddings to find topics (Sia et al., 2020), we want to use clusters of pre-trained sentence embeddings to represent the semantic centers of the text. Given a text, We use Sentence BERT model[2] (Reimers and Gurevych, 2019) to get sentence embeddings and then use K-means to compute two clusters. Semantic coherence is measured by the cosine similarity between the two clusters. The more similar the clusters are, the more coherent the text is.

Testcase	TicketTalk	TM2
BB	0.5279	0.6973
Forced 2 BB	0.5512	0.6862
Forced 3 BB	0.5579	0.6645
Random	0.5771	0.6625
Truth	0.6095	0.5804

Table 1: Cosine Similarities Between Clusters of Sentence Embeddings. (Higher score means higher similarity.)

Texts generated by the following five models are evaluated,

- Normal Brownian bridge model (BB),
- Brownian bridge model with expanded sampling region to twice (Forced 2 BB),
- Brownian bridge model with expanded sampling region to third (Forced 3 BB),
- Random sampling model (Random),
- Ground truth model (Truth).

The results of cosine similarities are reported in Table 1. While the Brownian bridge model has better semantic coherence on TM2 dataset, the random sampling model performs better on the TicketTalk dataset.

4.4.2 Automatic topic coherence

In Table 2, we show the topic coherence evaluation of the model trained with TicketTalk dataset and in Table 3, we show the result trained with TM2 dataset.

In these result, the higher the score is, the more coherent the model is. Therefore, as we can see in

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Testcase	C_{UCI}	C_{NPMI}	C_{UMass}
BB	0.4033	0.0188	-26.30
Forced 2 BB	0.4034	0.0181	-26.69
Forced 3 BB	0.3619	0.0175	-26.56
Random	0.9692	0.0366	-26.44
Truth	0.8109	0.0323	-26.55

Table 2: Topic coherence evaluation, TicketTalk

Testcase	C_{UCI}	C_{NPMI}	C_{UMass}
BB	6.303	0.2342	-20.95
Forced 2 BB	4.653	0.1758	-22.45
Forced 3 BB	3.410	0.1271	-23.87
Random	7.276	0.2674	-20.03
Truth	3.558	0.1290	-24.04

Table 3: Topic coherence evaluation, TM2

both these two topic, the random model and ground truth result is the most coherent one. For the rest Brownian bridge model, apparently, those forced model shows worse result compared with the normal Brownian bridge model, which corresponds with our expectation.

4.4.3 Human Evaluation

In order to test the efficiency of our evaluation metrics, we also conducted human evaluations on the same text. Three evaluators assess the coherence of the given text independently and assign a score from 0 to 10 for each document. Then the average score (1-10) was calculated for each experiment setting, as shown in Table 4. Figure 7 shows the UCI coherence correlates with our human evaluation in TicketTalk dataset. We observed similar correlation from TM2 dataset as well.

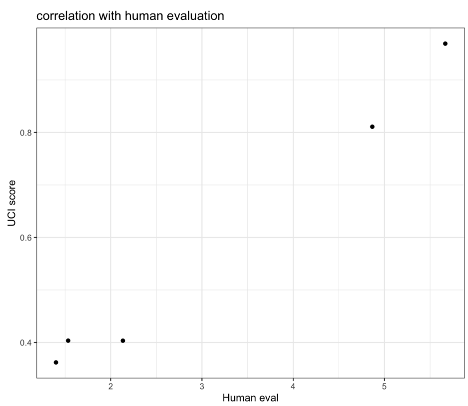


Figure 7: correlation with human

Testcase	TicketTalk	TM2
BB	2.13	1.47
Forced 2 BB	1.53	1.13
Forced 3 BB	1.40	1.00
Random	5.67	1.53
Truth	4.87	5.93

Table 4: Human Evaluation

5 Discussion

To sum up, in this project we study the latent space characterized by Brownian Bridge process and analyze its relationship with text coherence. We observed that all the dimension of the latent space have similar shape driven by the objective function, which is expected. Also, we found that lots of latent trajectories have big jump towards the end of the conversation, which is because the start and the end of a conversation are generally easier to identified while the signal from the middle of the conversations are not very strong. Interestingly, we also found that among the different embedding design, the random embeddings seem to perform well on the TicketTalk dataset, even better than the true embedding settings. However, this finding does not hold in the TM2 dataset. From our experiments, true embeddings do give good performances across different settings and pinned the trajectories at the tails of the distribution do destroy the coherence. Overall, we found employing the idea of stochastic process in learning the representation help us better understand how to generate the coherent text and achived the goal of anchoring the end target. With that being said, how to further improve this modeling approach and control the length, topics or other aspects for dialogue generation still worth investigation. Our code for this project can be found at Github.

6 Future Directions

Based on the preliminary results, there are still lots of remaining questions to be answered. For example, we are still looking for a suitable statistical inference tool to test the similarity between a random generated stochastic sequence with the average Brownian bridge trajectory calculated from the true parameters. Additionally, we are going to further study the mechanism of the decoder from the original paper because the model seems to be generating inconsistent length of text when tuning the variance of the Brownian bridge even if we set it upfront. We suspects this behavior is related to the decoder.

Specifically, we would like to further eliminate the effect from the decoder, and replace the decoder part by some other transformer models such as *DialoGPT* and *DialoFlow*. Moreover, we would also like to conduct larger scale of experiments and find ways to quantify coherence for the generated text. Currently we are using human evaluation but hopefully can utilize some automatic evaluation metrics such as perplexity to help increase the scalability. Furthermore, we can also design some handcraft metric like ordering, absence, redundancy and topic-diversifying cut point to evaluate the coherence of generated text.

Ultimately, we are hoping to add more features such flow index, external knowledge such as emotion and personality into the learned latent embeddings to get more control over the generation procedure. However, it almost surely will surpass the scope of this semester.

7 Broader Impact

Distinguished from task-oriented dialogue system, open-domain dialogue system aims to establish long-term connections with users to satisfy the need for social belongings or communications(Huang et al., 2019). Under certain conditions, open-domain conversational agent is also expected to solve certain task. Through this project, we hope to develop a method that can generate consistent and coherent long dialogues from limited prior knowledge. Ideally, the product out of the project can help writers gain more ideas from machine generated dialogues and help developers to build more versatile chat robots. Last but not the least, it can also help generate humongous training materials to better train other types of language models.

References

Rilwan A. Adewoyin, Ritabrata Dutta, and Yulan He. 2022. Rstgen: Imbuing fine-grained interpretable control into long-formtext generators.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2 – how can i help you? towards the use of pretrained language models for task-oriented dialogue systems.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A novel neural topic model and its supervised extension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12911–12919.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2019. Challenges in building intelligent open-domain dialog systems.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Jingyuan Li and Xiao Sun. 2018. A syntactically constrained bidirectional-asynchronous approach for emotional conversation generation. *arXiv preprint arXiv:1806.07000*.

Juntao Li, Chang Liu, Chongyang Tao, Zhangming Chan, Dongyan Zhao, Min Zhang, and Rui Yan. 2021a. Dialogue history matters! personalized response selection in multi-turn retrieval-based chatbots. *ACM Trans. Inf. Syst.*, 39(4).

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021b. Conversations are not flat: Modeling the dynamic information flow across dialogue utterances. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 128–138, Online. Association for Computational Linguistics.

Yuanxin Liu, Zheng Lin, Fenglin Liu, Qinyun Dai, and Weiping Wang. 2019. Generating paraphrase with topic as prior knowledge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2381–2384.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *International Conference on Machine Learning*, pages 2410–2419. PMLR.

Allen Nie, Erin Bennett, and Noah Goodman. 2019. DisSent: Learning sentence representations from explicit discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4497–4510, Florence, Italy. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736, Online. Association for Computational Linguistics.

Sifatullah Siddiqi and Aditi Sharan. 2015. Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications*, 109(2).

Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019a. Exploiting persona information for diverse generation of conversational responses. *arXiv preprint arXiv:1905.12188*.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuan-Jing Huang. 2019b. Generating responses with a specific emotion in dialog. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3695.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. 2022. Language modeling via stochastic processes.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Minghong Xu, Piji Li, Haoran Yang, Pengjie Ren, Zhaochun Ren, Zhumin Chen, and Jun Ma. 2020. A neural topical expansion framework for unstructured persona-oriented dialogue generation. *arXiv preprint arXiv:2002.02153*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys (CSUR)*.

Jian Zhang, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022. Less is more: Learning to refine dialogue history for personalized dialogue generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5808–5820, Seattle, United States. Association for Computational Linguistics.

Appendix

Extreme trajectory

As shown in Figure 8, the trajectory is manually selected to be boundary of the initial sampling region.

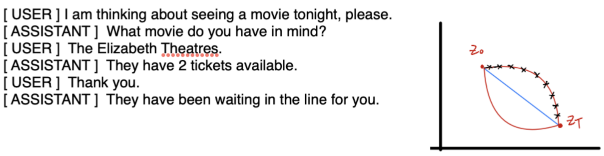


Figure 8: Extreme Trajectory: case 1

As shown in Figure 9, the trajectory is manually selected to be boundary of the sampling region which is expanded to twice the original region.

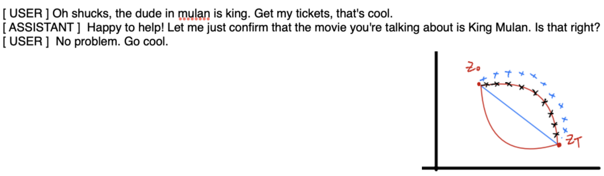


Figure 9: Extreme Trajectory: case 2

As shown in Figure 10, the trajectory is manually selected to be boundary of the sampling region which is expanded to three times the original region.

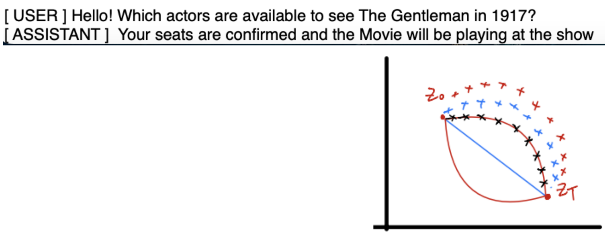


Figure 10: Extreme Trajectory: case 3

Extended sampling region

As shown in Figure 11, the size of sampling region is the same as the initial region used in trained model:

As shown in Figure 12, the sampling region is expanded to twice the original region.

As shown in Figure 13, the sampling region is expanded to three times the original region.

For evaluation, we added another artificial agent as our annotator and collected its assessments for our experiment materials. Figure 14 shows an instance of how to design the prompt and the corresponding response from ChatGPT.

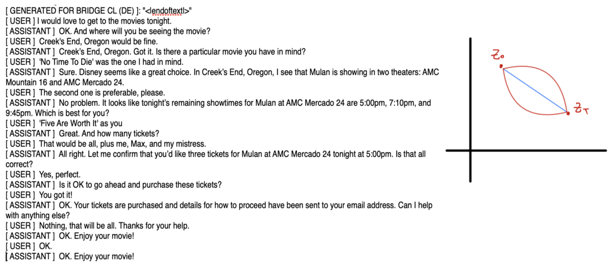


Figure 11: Extended sampling region: case 1

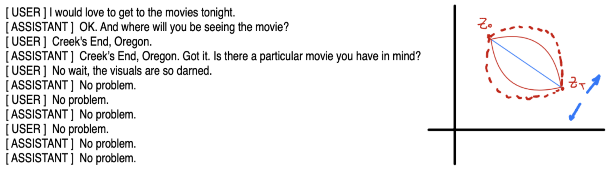


Figure 12: Extended sampling region: case 2

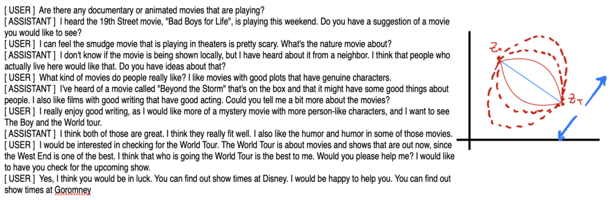


Figure 13: Extended sampling region: case 3

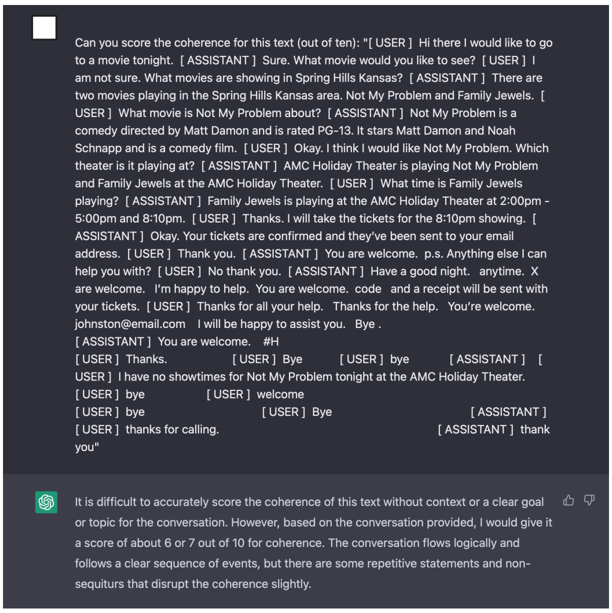


Figure 14: ChatGPT screenshot