

A Sentiment-Controllable Topic-to-Essay Generator with Topic Knowledge Graph

Lin Qiao^{1*}, Jianhao Yan², Fandong Meng², Zhendong Yang, and Jie Zhou²

¹School of Software and Microelectronics, Peking University

²Pattern Recognition Center, WeChat AI, Tencent Inc, Beijing, China

qiaolin66666@gmail.com

{elliottyan,fandongmeng,withtomzhou}@tencent.com

Abstract

Generating a vivid, novel, and diverse essay with only several given topic words is a challenging task of natural language generation. In previous work, there are two problems left unsolved: neglect of sentiment beneath the text and insufficient utilization of topic-related knowledge. Therefore, we propose a novel Sentiment-Controllable topic-to-essay generator with a Topic Knowledge Graph enhanced decoder, named SCTKG, which is based on the conditional variational auto-encoder (CVAE) framework. We firstly inject the sentiment information into the generator for controlling sentiment for each sentence, which leads to various generated essays. Then we design a Topic Knowledge Graph enhanced decoder. Unlike existing models that use knowledge entities separately, our model treats knowledge graph as a whole and encodes more structured, connected semantic information in the graph to generate a more relevant essay. Experimental results show that our SCTKG can generate sentiment controllable essays and outperform the state-of-the-art approach in terms of topic relevance, fluency, and diversity on both automatic and human evaluation.

1 Introduction

Topic-to-essay generation (TEG) task aims at generating human-like paragraph-level texts with only several given topics. It has plenty of practical applications, e.g., automatic advertisement generation, intelligent education, or assisting in keyword-based news writing (Leppänen et al., 2017). Because of its great potential in practical use and scientific research, TEG has attracted a lot of interest. (Feng et al., 2018; Yang et al., 2019). However, In TEG, two problems are left to be solved: the neglect of sentiment beneath the text and the insufficient utilization of topic-related knowledge.

*This work is done when Lin Qiao was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China

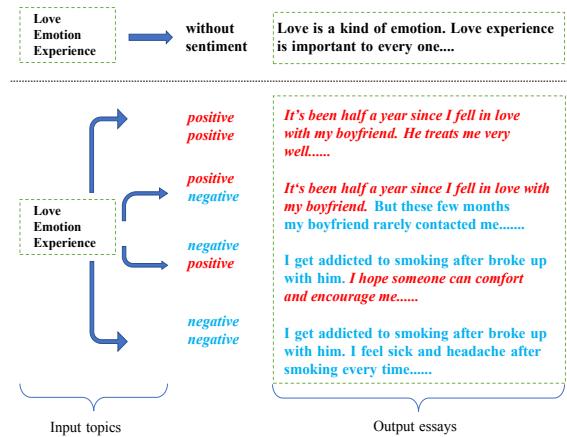


Figure 1: Examples of comparison between the generated essays with sentiment control and without sentiment. We show the first two sentences for each generated essay and denote positive sentences in red and negative sentences in blue. Sentences without sentiment label are showed in black.

A well-performed essay generator should be able to generate multiple vivid and diverse essays when given the topic words. However, previous work tends to generate dull and generic texts. One of the reason is that they neglect the sentiment factor of the text. By modeling and controlling the sentiment of generated sentences, we can generate much more diverse and fascinating essays. As shown in Figure 1, given the topic words “Love”, “Experience” and “Emotion”, the “without sentiment” model generates monotonous article. In contrast, the sentiment-attach model generates positive statements such as “fall in love with my boyfriend” when given the “positive” label, and generates negative phrases such as “addicted to smoking”, “broke up” when given the “negative” label. In addition, sentiment control is especially essential in topic-to-essay generation task, which aims to generate multiple sentences. As the number of sentences increases, the search space for generation model is exponentially enlarged by controlling the sentiment

polarity for each of the sentence. Therefore, the ability to control sentiment is essential to improve discourse-level diversity for the TEG task.

As for the other problem, imagine that when we human beings are asked to write articles with some topics, we heavily rely on our commonsense knowledge related to the topics. Therefore, the proper usage of knowledge plays a vital role in the topic-to-essay generation. Previous state-of-the-art method (Yang et al., 2019) extracts topic-related concepts from a commonsense knowledge base to enrich the input information. However, they ignore the graph structure of the knowledge base, which merely refer to the concepts in the knowledge graph and fail to consider their correlation. This limitation leads to concepts being isolated from each other. For instance, given two knowledge triples (*law*, *antonym*, *disorder*) and (*law*, *part of*, *theory*), about the topic word *law*, Yang et al. (2019) simply uses the neighboring concepts *disorder* and *theory* as a supplement to the input information. However, their method fails to learn that *disorder* has opposite meaning with *law* while *theory* is a hypernym to *law*, which can be learned from their edges (correlations) in the knowledge graph. Intuitively, lacking the correlation information between concepts in the knowledge graph hinders a model from generating appropriate and informative essays.

To address these issues, we propose a novel **Sentiment-Controllable** topic-to-essay generator with a **Topic Knowledge Graph** enhanced decoder, named SCTKG, which is based on the conditional variational auto-encoder (CVAE) framework. To control the sentiment of the text, we inject the sentiment information in the encoder and decoder of our model to control the sentiment from both sentence level and word level. The sentiment labels are provided by a sentiment classifier during training. To fully utilize the knowledge, the model retrieves a topic knowledge graph from a large-scale commonsense knowledge base ConceptNet (Speer and Havasi, 2012). Different from Yang et al. (2019), we preserve the graph structure of the knowledge base and propose a novel *Topic Graph Attention* (TGA) mechanism. TGA attentively reads the knowledge graphs and makes the full use of the structured, connected semantic information from the graphs for a better generation. In the meantime, to make the generated essays more closely surround the semantics of all input topics, we adopt adversarial training based on a multi-label

discriminator. The discriminator provides the reward to the generator based on the coverage of the output on the given topics.

Our contributions can be summarized as follow:

1. We propose a sentiment-controllable topic-to-essay generator based on CVAE, which can generate high-quality essays as well as control the sentiment. To the best of our knowledge, we are the first to control the sentiment in TEG and demonstrate the potential of our model to generate diverse essays by controlling the sentiment.
2. We equip our decoder with a topic knowledge graph and propose a novel Topic Graph Attention (TGA) mechanism. TGA makes the full use of the structured, connected semantic information from the topic knowledge graph to generate more appropriate and informative essays.
3. We conduct extensive experiments, showing that our model accurately controls the sentiment and outperforms the state-of-the-art methods both in automatic and human evaluations.

2 Task Formulation

Traditional TEG task takes as input a topic sequence $X = (x_1, \dots, x_m)$ with m words, and aims to generate an essay with M sentences (L_1, \dots, L_M) corresponding with topic sequence X . In this paper, we provide a sentiment sequence $S = (s_1, \dots, s_M)$, each of which corresponds to a target sentence in essay. Each sentiment can be *positive*, *negative*, or *neutral*.

Essays are generated in a sentence-by-sentence manner. The first sentence L_1 is generated only conditioned on the topic sequence X , then the model takes all the previous generated sentences as well as the topic sequence to generate the next sentence until the entire essay is completed. In this paper, we denote the previous sentences $L_{1:i-1}$ as context.

3 Model Description

In this section, we describe an overview of our proposed model. Our SCTKG generator based on a CVAE architecture consists of an encoder and a topic knowledge graph enhanced decoder. The encoder encodes topic sequence, sentiment, and

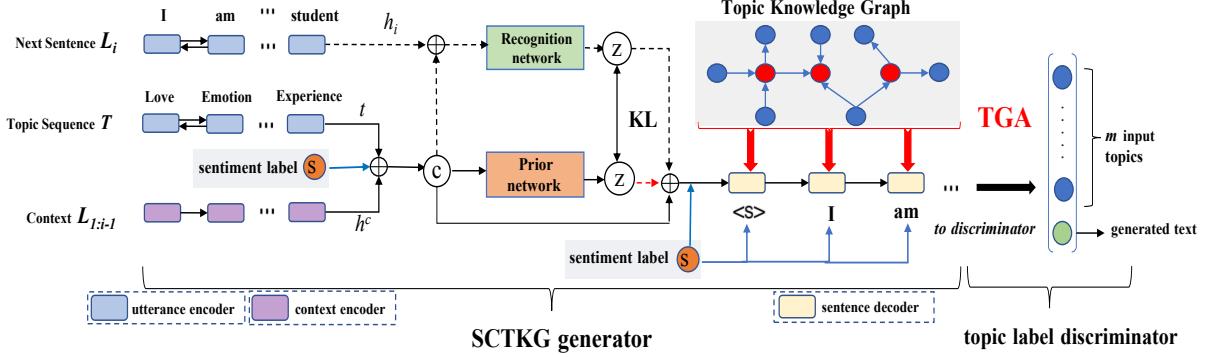


Figure 2: The architecture of our model. \oplus denotes the vector concatenation operation. Only the part with solid lines and the red dotted arrow is applied at inference, while the entire CVAE except the red dotted arrow part used in the training process. Sentiment label s with blue arrows denote sentiment control. Red solid lines denote TGA at each decoding step. The text generated by SCTKG generator feeds to topic label discriminator. The above m blue circle represents the probability that it belongs to the real text with the m input topics, and the green circle represents the given text is a generated text.

context and regards them as conditional variables c . Then a latent variable z is computed from c through a recognition network (during training) or prior network (during inference). The decoder attaches with a topic knowledge graph and sentiment label to generate the texts. At each decoding step, the TGA is used to enrich input topic information through effectively utilizing the topic knowledge graph.

We adopt a two-stage training approach: (1) Train the SCTKG generator with the conventional CVAE loss; (2) After the first step is done, we introduce a topic label discriminator to evaluate the performance of SCTKG generator. We adopt adversarial training to alternately train the generator and the discriminator to further enhance the performance of the SCTKG generator.

3.1 SCTKG Generator

3.1.1 Encoder

As shown in Figure 2, the utterance encoder is a bidirectional GRU (Chung et al., 2014) to encode an input sequence into a fixed-size vector by concatenating the last hidden states of the forward and backward GRU. We use the utterance encoder to encode the topic sequence X into $h^x = [\vec{h}^x, \overleftarrow{h}^x]$, $h^x \in \mathbb{R}^d$. d is the dimension of the vector. The next sequence L_i is also encoded by utterance encoder into $h_i = [\vec{h}_i, \overleftarrow{h}_i]$, $h_i \in \mathbb{R}^d$. For context encoder, we use a hierarchical encoding strategy. Firstly, each sentence in context $L_{1:i-1}$ is encoded by utterance encoder to get a fixed-size vector. By doing so, the context $L_{1:i-1}$ is encoded

into $h_{context} = [h_1, h_2, \dots, h_{i-1}]$. Then a single layer forward GRU is used to encode the sentence representations $h_{context}$ into a final state vector $h^c \in \mathbb{R}^d$.

Then the concatenation of h^c , h^x , $e(s)$ is functionalized as the conditional vector $c = [e(s); h^c; h^x]$. $e(s)$ is the embedding of sentiment label s . We assume that z follows a multivariate Gaussian distribution with a diagonal covariance matrix. Thus the recognition network $q_\phi(z|h_i, c)$ and the prior network $p_\theta(z|c)$ follow $\mathcal{N}(\mu, \sigma^2 \mathbf{I})$ and $\mathcal{N}(\mu', \sigma'^2 \mathbf{I})$, respectively. I is identity matrix, and then we have

$$\begin{aligned} [\mu, \sigma^2] &= \text{MLP}_{\text{recognition}}(h_i, c), \\ [\mu', \sigma'^2] &= \text{MLP}_{\text{prior}}(c). \end{aligned} \quad (1)$$

Additionally, we use a reparametrization trick (Kingma and Welling, 2013) to sample z from the recognition network during training and from prior network during testing.

3.1.2 Decoder

A general Seq2seq model may tend to emit generic and meaningless sentences. To create more meaningful essays, we propose a topic knowledge graph enhanced decoder. The decoder is based on a 1-layer GRU network with initial state $d_0 = W_d[z, c, e(s)] + b_d$. W_d and b_d are trainable decoder parameters and $e(s)$ is the sentiment embedding as mentioned above. As shown in Figure 2, we equip the decoder with a topic knowledge graph to incorporate commonsense knowledge from ConceptNet¹. ConceptNet is a semantic network which

¹<https://conceptnet.io>

consists of triples $R = (\text{head}; \text{rel}; \text{tail})$. The head concept *head* has the relation *rel* with tail concept *tail*. We use word vectors to represent head and tail concepts and learn trainable vector \mathbf{r} for relation *rel*, which is randomly initialized. Each word in the topic sequence is used as a query to retrieve a subgraph from ConceptNet and the topic knowledge graph is constituted by these subgraphs. Then we use the Topic Graph Attention (TGA) mechanism to read from the topic knowledge graph at each generation step.

Topic Graph Attention. As previously stated, a proper usage of the external knowledge plays a vital role in our task. TGA takes as input the retrieved topic knowledge graph and a query vector \mathbf{q} to produce a graph vector g_t . We set $\mathbf{q} = [d_{t-1}; c; z]$, where d_{t-1} represents the decoder hidden state for $t - 1$ step. At each decoding step, we calculate the correlation score between each of the triples in the graph and \mathbf{q} . Then we use the correlation score to compute the weighted sum of all the neighboring concepts² to the topic words to form the final graph vector g_t . Neighboring concepts are entities that directly link to topic words. We formalize the computational process as follows:

$$g_t = \sum_{n=1}^N \alpha_n \mathbf{o}_n, \quad (2)$$

$$\alpha_n = \frac{\exp(\beta_n)}{\sum_{j=1}^N \exp(\beta_j)}, \quad (3)$$

$$\beta_n = \begin{cases} (\mathbf{W}_1 \mathbf{q})^\top \tanh(\mathbf{W}_2 \mathbf{r}_n + \mathbf{W}_3 \mathbf{o}_n) & \text{when } \mathbf{o}_n \in \mathcal{S}_1 \\ (\mathbf{W}_1 \mathbf{q})^\top \tanh(\mathbf{W}_2 \mathbf{r}_n + \mathbf{W}_4 \mathbf{o}_n) & \text{when } \mathbf{o}_n \in \mathcal{S}_2 \end{cases}, \quad (4)$$

where \mathbf{o}_n is the embedding of n^{th} neighboring concept and \mathbf{r}_n is the embedding of the relation for n^{th} triple in the topic knowledge graph. $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4$ are weight matrices for query, relations, head entities and tail entities, respectively. \mathcal{S}_1 contains the neighboring concepts which being the head concepts in their triples, while \mathcal{S}_2 contains the neighboring concepts which being the tail concepts. The matching score β_n represents the correlation between the query \mathbf{q} and neighbouring concept \mathbf{o}_n . Essentially, a graph vector g_t is the

²As shown in Figure 2, in the topic knowledge graph, red circles denote the topic words and blue circles denote their neighboring concepts. Since we have already encoded topic information in the encoder, the graph vector g_t in this section mainly focuses on the neighboring concept to assist the generation.

weighted sum of the neighbouring concepts of the topic words. Note that we use different weight matrices to distinguish the neighboring concepts in different positions (in head or in tail). This distinction is necessary. For instance, given two knowledge triples (*Big Ben*, *part of*, *London*) and (*London*, *part of*, *England*). Even though the concepts *Big Ben* and *England* are both neighboring concepts to *London* with the same relation *part of*, they have the different meaning with regard to *London*. We need to model this difference by \mathbf{W}_3 and \mathbf{W}_4 .

Then the final probability of generating a word is computed by

$$P_t = \text{softmax}(W_o[d_t; e(s); g_t] + b_o),$$

where d_t is the decoder state at t step and $W_o \in \mathbb{R}^{d_{\text{model}} \times |V|}$, $b_o \in \mathbb{R}^{|V|}$ are trainable decoder parameters. d_{model} is the dimension of $[d_t; e(s); g_t]$ and $|V|$ is vocabulary size.

3.2 Topic Label Discriminator

Another concern is that the generated texts should be closely related to the topic words. To this end, at the second training stage, a topic label discriminator is introduced to perform adversarial training with the SCTKG generator. In a max-min game, the SCTKG generator generates essays to make discriminator consider them semantically match with given topics. Discriminator tries to distinguish the generated essays from real essays. In detail, suppose there are a total of m topics, the discriminator produces a sigmoid probability distribution over $(m + 1)^{th}$ classes. The score at $(m + 1)^{th}$ index represents the probability that the sample is the generated text. The score at the j^{th} ($i \in \{1, \dots, m\}$) index represents the probability that it belongs to the real text with the j^{th} topic. Here the discriminator is a CNN (Kim, 2014) text classifier.

3.3 Training

We introduce our two stage training method in this section. Stage 1: Similar to a conventional CVAE model, The loss of our SCTKG generator $-\log p(Y|c)$ can be expressed as:

$$\begin{aligned} -\mathcal{L}(\theta; \phi; c; Y)_{\text{cvae}} &= \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{decoder}} \\ &= \text{KL}(q_\phi(z|Y, c) \| p_\theta(z|c)) \\ &\quad - \mathbb{E}_{q_\phi(z|Y, c)} (\log p_D(Y|z, c)). \end{aligned} \quad (5)$$

Here, θ and ϕ are the parameters of the prior network and recognition network, respectively. Intuitively, $\mathcal{L}_{\text{decoder}}$ maximizes the sentence generation probability after sampling from the recognition net-

work, while \mathcal{L}_{KL} minimizes the distance between the prior and recognition network. Besides, we use the annealing trick and BOW-loss (Zhao et al., 2017) to alleviate the vanishing latent variable problem in VAE training.

Stage 2: After trained the SCTKG generator with equation (5), inspired by SeqGan (Yu et al., 2017), we adopt adversarial training between the generator and the topic label discriminator described in section 3.2. We refer reader to Yu et al. (2017) and Yang et al. (2019) for more details.

4 Experiments

4.1 Datasets

We conduct experiments on the ZHIHU corpus (Feng et al., 2018). It consists of Chinese essays³ whose length is between 50 and 100. We select topic words based on frequency and remove rare topic words. The total number of topic labels are set to 100. Sizes of the training set and the test set are 27,000 essays and 2500 essays. For tuning hyperparameters, we set aside 10% of training samples as the validation set.

The sentence sentiment labels is required for our model training. To this end, we sample 5000 sentences from the dataset and annotated the data manually with three categories, i.e., positive, negative, neutral. This dataset was divided into a training set, validation set, and test set. We use an open-source Chinese sentiment classifier Senta⁴ to finetune on our manually-label training set. This classifier achieves an accuracy of 0.83 on the test set. During training, the target sentiment labels s is computed by the sentiment classifier automatically. During inference, users can input any sentiment labels to control the sentiment for sentence generation.

4.2 Implementation Details

We use the 200-dim pre-trained word embeddings provided by Song et al. (2018) and dimension of sentiment embeddings is 32. The vocabulary size is 50,000 and the batch size is 64. We use a manually tuning method to choose the hyperparameter values and the criterion used to select is BLEU (Papineni et al., 2002a). We use GRU with hidden size 512 for both encoder and decoder and the size of latent variables is 300. We implement the model with

³The dataset can be download by <https://pan.baidu.com/s/17pcfWUuQTbcbniT0tBdwFQ>

⁴<https://github.com/baidu/Senta>

Tensorflow⁵. The number of parameters is 68M and parameters of our model were randomly initialized over a uniform distribution [-0.08,0.08]. We pre-train our model for 80 epochs with the MLE method and adversarial training for 30 epochs. The average runtime for our model is 30 hours on a Tesla P40 GPU machine, which adversarial training takes most of the runtime. The optimizer is Adam (Kingma and Ba, 2014) with 10^{-3} learning rate for pre-training and 10^{-5} for adversarial training. Besides, we apply dropout on the output layer to avoid over-fitting (Srivastava et al., 2014) (dropout rate = 0.2) and clip the gradients to the maximum norm of 10. The decoding strategy in this paper uses greedy search and average length of generated essays is 79.3.

4.3 Evaluation

To comprehensively evaluate the generated essays, we rely on a combination of both automatic evaluation and human evaluation.

Automatic Evaluation. Following previous work (Yang et al., 2019), we consider the following metrics⁶:

BLEU: The BLEU score (Papineni et al., 2002b) is widely used in machine translation, dialogue, and other text generation tasks by measuring word overlapping between ground truth and generated sentences.

Dist-1, Dist-2 (Li et al., 2015): We calculate the proportion of distinct 1-grams and 2-grams in the generated essays to evaluate the diversity of the outputs.

Consistency (Yang et al., 2019): An ideal essay should closely surround the semantics of all input topics. Therefore, we pre-train a multi-label classifier to evaluate the topic-consistency of the output. A higher “Consistency” score means the generated essays are more closely related to the given topics.

Novelty (Yang et al., 2019): We calculated the novelty by the difference between output and essays with similar topics in the training corpus. A higher “Novelty” score means the output essays are more different from essays in the training corpus.

Precision, Recall and Senti-F1: These metrics are used to measure sentiment control accuracy. If the sentiment label of the generated sentence is consistent with the ground truth, the generated result

⁵<https://github.com/tensorflow/tensorflow>

⁶<https://github.com/libing125/CTEG>

Methods	Automatic evaluation					Human evaluation			
	BLEU	Consistency	Novelty	Dist-1	Dist-2	Con.	Nov.	E-div.	Flu.
TAV	6.05	16.59	70.32	2.69	14.25	2.32	2.19	2.58	2.76
TAT	6.32	9.19	68.77	2.25	12.17	1.76	2.07	2.32	2.93
MTA	7.09	25.73	70.68	2.24	11.70	3.14	2.87	2.17	3.25
CTEG	9.72	39.42	75.71	5.19	20.49	3.74	3.34	3.08	3.59
SCTKG(w/o-Senti)	9.97	43.84	78.32	5.73	23.16	3.89	3.35	3.90	3.71
SCTKG(Ran-Senti)	9.64	41.89	79.54	5.84	23.10	3.80	3.48	4.29	3.67
SCTKG(Gold-Senti)	11.02	42.57	78.87	5.92	23.07	3.81	3.37	3.94	3.75

Table 1: Automatic and human evaluation result. In human evaluation, **Con.**, **Nov.**, **E-div.**, **Flu.** represent topic-consistency, novelty, essay-diversity, fluency, respectively. The best performance is highlighted in bold.

is right, and wrong otherwise. The sentiment label is predicted by our sentiment classifier mentioned above (see 4.1 for details about this classifier).

Human Evaluation. We also perform human evaluation to more accurately evaluate the quality of generated essays. Each item contains the input topics and outputs of different models. Then, 200 items are distributed to 3 annotators, who have no knowledge in advance about the generated essays come from which model. Each annotator scores 200 items and we average the score from three annotators. They are required to score the generated essay from 1 to 5 in terms of three criteria: **Novelty**, **Fluency**, and **Topic-Consistency**. For novelty, we use the TF-IDF features of topic words to retrieve 10 most similar training samples to provide references for the annotators. To demonstrate the paragraph-level diversity of our model, we propose a **Essay-Diversity** criteria. Specifically, each model generates three essays with the same input topics, and annotators are required to score the diversity by considering the three essays together.

4.4 Baselines

TAV (Feng et al., 2018) represents topic semantics as the average of all topic embeddings and then uses a LSTM to generate each word. Their work also includes the following two baselines.

TAT (Feng et al., 2018) extends LSTM with an attention mechanism to model the semantic relatedness of each topic word with the generator’s output.

MTA (Feng et al., 2018) maintains a topic coverage vector to guarantee that all topic information is expressed during generation through an LSTM decoder.

CTEG (Yang et al., 2019) adopts commonsense knowledge and adversarial training to improve gen-

eration. It achieves state-of-the-art performance on the topic-to-essay generation task.

5 Results and Analysis

In this section, we introduce our experimental results and analysis from two part: the “text quality” and “sentiment control”. Then we show case study of our model.

5.1 Results on Text Quality

The automatic and human evaluation results are shown in Table 1. We present three different versions of our model for a comprehensive comparison. (1)“SCTKG(w/o-Senti)” means we do not attach any sentiment label to the model. (2) “SCTKG(Ran-Senti)” means we randomly set the sentiment label for each generated sentence. (3) “SCTKG(Gold-Senti)” means we set the golden sentiment label for the generated sentence. By investigating the results in Table 1, we have the following observations:

First, all versions of our SCTKG models outperform the baselines in all evaluation metrics (except the BLEU score of SCTKG(Ran-Senti)). This demonstrates that our SCTKG model can generate better essays than baseline models, whether uses the true sentiment, random sentiment or without any sentiment.

Second, we can learn the superiority of the basic architecture of our model through the comparison between SCTKG(w/o-Senti) and the baselines. In human evaluation, SCTKG(w/o-Senti) outperform CTEG in topic-consistency, essay-diversity, and fluency by +0.15 (3.74 vs 3.89), +0.82 (3.08 vs 3.90), +0.12 (3.59 vs 3.71) respectively. Similar improvements can be also drawn from the automatic evaluation. The improvement in essay-diversity

is the most significant. This improvement comes from our CVAE architecture because our sentence representation comes from the sampling from a continuous latent variable. This sampling operation introduces more randomness compared with baselines.

Third, as previously stated, each model generates three essays and considers them as a whole when comparing the “E-div”. When given the random and diverse sentiment label sequences, our SCTKG(Ran-Senti) achieves the highest “E-div” score (4.29). Consider that CVAE architecture has already improved the diversity compared with baselines. By randomizing the sentiment of each sentence, SCTKG(Ran-Senti) further boosts this improvement (from +0.82 to +1.21 compared with CTEG). This result demonstrates the potential of our model to generate discourse-level diverse essays by using diverse sentiment sequences, proving our claim in the introduction part.

Fourth, when using the golden sentiment label, SCTKG(Gold-Senti) achieves the best performance in BLEU (11.02). However, we find the SCTKG(Gold-Senti) do not significantly outperforms other SCTKG models in other metrics. The results show the true sentiment label of the target sentence benefits SCTKG(Gold-Senti) to better fit in the test set, but there is no obvious help for other important metrics such as diversity and topic-consistency.

Fifth, we find it interesting that when removing the sentiment label, the SCTKG(w/o-Senti) achieves the best topic-consistency score. We conceive that sentiment label may interfere with the topic information in the latent variable to some extent. But the effect of this interference is trivial. Comparing SCTKG(w/o-Senti) and SCTKG(Gold-Senti), the topic-consistency only drops 0.08 (3.89 vs 3.81) for human evaluation and 1.27 (43.84 vs 42.57) for automatic evaluation, which is completely acceptable for a sentiment controllable model.

Ablation study on text quality. To understand how each component of our model contributes to the task, we train two ablated versions of our model: without adversarial training (“w/o AT”) and without TGA (“w/o TGA”). Noted that in the “w/o TGA” experiment, we implement a memory network the same as [Yang et al. \(2019\)](#) which uses the concepts in ConceptNet but regardless of their correlation. All models uses golden sentiment la-

Methods	BLEU	Con.	Nov.	E-div.	Flu.
Full model	11.02	3.81	3.37	3.94	3.75
w/o TGA	10.34	3.54	3.17	3.89	3.38
w/o AT	9.85	3.37	3.20	3.92	3.51

Table 2: Ablation study on text quality. “w/o AT” means without adversarial training. “w/o TGA” means without TGA. **Con., Nov., E-div., Flu.** represent topic-consistency, novelty, essay-diversity, fluency, respectively. Full model represent SCTKG(Gold-Senti) in this table.

bel. Table 2 presents the BLEU scores and human evaluation results of the ablation study.

By comparing full model and “w/o TGA”, we find that without TGA, the model performance drops in all metrics. In particularly, topic-consistency drops 0.27, which shows that by directly learning the correlation between the topic words and its neighboring concepts, concepts that are more closely related to the topic words are given higher attention during generation. Novelty drops 0.2, the reason is that TGA is an expansion of the external knowledge graph information. Therefore the output essays are more novel and informative. Fluency drops 0.37 because TGA benefits our model to choose a more suitable concept in the topic knowledge graph according to the current context. And the BLEU drops for 0.68 shows TGA helps our model to better fit the dataset by modeling the relations between topic words and neighboring concepts.

By comparing full model and “w/o AT”, we find that adversarial training can improve the BLEU, topic-consistency, and fluency. The reason is that the discriminative signal enhancing the topic consistency and authenticity of the generated texts.

5.2 Results on Sentiment Control

In this section, we investigate whether the model accurately control the sentiment and how each component affects our sentiment control performance. We train three ablated versions of our model: without sentiment label in encoder, without sentiment label in decoder, and without TGA. We randomly sample 50 essays in our test set with 250 sentences. Instead of using golden sentiment labels, the sentiment labels are randomly given in this section. Predicting the golden sentiment is relatively simple because sometimes emotional labels can be directly derived from the coherence between contexts. We adopt a

Methods	Precison	Recall	Senti-F1
Full model	0.68	0.66	0.67
w/o Enc-senti	0.56	0.55	0.56
w/o Dec-senti	0.59	0.62	0.61
w/o TGA	0.62	0.64	0.63

Table 3: Ablation study on sentiment control. “w/o Enc-senti” means to remove the sentiment embedding in the encoder side and “w/o Dec-senti” means to remove from the decoder. Full model represents SCTKG(Ran-Senti) in this table.

Input topics: Law Education
Sentiment label: neu. pos. neg. neg. neu.
Output essay: I am a senior high school student. I am in the best high school in our town. But bullying still exist on our campus. Teachers always ignore this phenomenon. What should we do to protect our rights?

Table 4: Given topic “Law” and “Education”, and randomly set sentiment label for each sentence. We generated an essay according to the topic and sentiment labels. “neu.” represents neutral. “pos.” represents positive and “neg.” represents negative. We have translated the original Chinese output into English.

more difficult experimental setting that aims to generate sentences following arbitrary given sentiment labels. The results are shown in Table 3.

We can learn that removing the sentiment label either from encoder or decoder leads to an obvious control performance decrease (-11% / -6% on Senti-F1) and the sentiment label in the encoder is the most important, since removing it leads to the most obvious decline (-11% Senti-F1). Although TGA does not directly impose sentiment information, it still helps to improve the control ability (4% in Senti-F1), which shows that learning correlations among concepts in topic knowledge graph strengthens the emotional control ability of the model. For instance, when given a positive label, the concepts related to the relation “desire of” are more likely to attach more attention, because the concepts with the relation “desire of” may represent more positive meaning.

5.3 Case Study

Table 4 presents an example of our output essay with a random sentiment sequence. Positive sentences are shown in red and negative sentences are shown in blue. We can learn that the output es-

say is not only closely related to the topic “Law” and “Education”, but also corresponding with the randomly given sentiment label. Meanwhile, our model makes full use of commonsense knowledge with the help of TGA. For example, “high school student” and “right” are the neighboring concepts related to the topic words “Education” and “Law”.

6 Related Work

Topic-to-Text Generation. Automatically generating an article is a challenging task in natural language processing. Feng et al. (2018) are the first to propose the TEG task and they utilize coverage vector to integrate topic information. Yang et al. (2019) use extra commonsense knowledge to enrich the input information and adopt adversarial training to enhancing topic-consistency. However, both of them fail to consider the sentiment factor in the essay generation and fully utilize the external knowledge base. These limitations hinder them from generating high-quality texts.

Besides, Chinese poetry generation is similar to our task, which can also be regarded as a topic-to-sequence learning task. Li et al. (2018) adopt CVAE and adversarial training to generate diverse poetry. Yang et al. (2017) use CVAE with hybrid decoders to generate Chinese poems. And Yi et al. (2018) use reinforcement learning to directly improve the diversity criteria. However, their models are not directly applicable to TEG task. Because they do not take knowledge into account, their models cannot generate long and meaningful unstructured essays.

Controllable Text Generation. Some work has explored style control mechanisms for text generation tasks. For example, Zhou and Wang (2017) use naturally annotated emoji Twitter data for emotional response generation. Wang and Wan (2018) propose adversarial training to control the sentiment of the texts. Chen et al. (2019) propose a semi-supervised CVAE to generate poetry and deduce a different lower bound to capture generalized sentiment-related semantics. Different from their work, we inject sentiment label in both encoder and decoder of CVAE and prove that by modeling a topic knowledge graph can further enhance the sentiment control ability.

7 Conclusions

In this paper, we make a further step in a challenging topic-to-essay generation task by proposing a

novel sentiment-controllable topic-to-essay generator with a topic knowledge graph enhanced decoder, named SCTKG. To get better representation from external knowledge, we present TGA, a novel topic knowledge graph representation mechanism. Experiments show that our model can not only generate sentiment-controllable essays but also outperform competitive baselines in text quality.

References

- Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, and Zhipeng Guo. 2019. *Sentiment-controllable chinese poetry generation*. pages 4925–4931.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-essay generation with neural networks. In *IJCAI*, pages 4078–4084.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Eprint Arxiv*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models.
- Juntao Li, Yan Song, Haisong Zhang, Dongmin Chen, Shuming Shi, Dongyan Zhao, and Rui Yan. 2018. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002a. *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.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002b. 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.
- Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 175–180.
- Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.
- Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2002–2012.
- Xiaopeng Yang, Xiaowen Lin, Shunda Suo, and Ming Li. 2017. Generating thematic chinese poetry using conditional variational autoencoders with hybrid decoders. *arXiv preprint arXiv:1711.07632*.
- Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.
- Xianda Zhou and William Yang Wang. 2017. Mojitalk: Generating emotional responses at scale. *arXiv preprint arXiv:1711.04090*.