Received: 24 March 2023 | Revised: 14 August 2023 | Accepted: 20 September 2023

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12280

ORIGINAL RESEARCH

A topic-controllable keywords-to-text generator with knowledge base network

Li He[1] | Kaize Shi[1] | Dingxian Wang[2] | Xianzhi Wang[1] | Guandong Xu[1]

[1]University of Technology Sydney, Broadway, Sydney, Australia

[2]Etsy.com, Seattle, Washington, USA

Correspondence

Li He.
Email: li.he-1@student.uts.edu.au

Funding information

Australian Research Council, Grant/Award Numbers: DP22010371, LE220100078

Abstract

With the introduction of more recent deep learning models such as encoder-decoder, text generation frameworks have gained a lot of popularity. In Natural Language Generation (NLG), controlling the information and style of the output produced is a crucial and challenging task. The purpose of this paper is to develop informative and controllable text using social media language by incorporating topic knowledge into a keyword-to-text framework. A novel Topic-Controllable Key-to-Text (TC-K2T) generator that focuses on the issues of ignoring unordered keywords and utilising subject-controlled information from previous research is presented. TC-K2T is built on the framework of conditional language encoders. In order to guide the model to produce an informative and controllable language, the generator first inputs unordered keywords and uses subjects to simulate prior human knowledge. Using an additional probability term, the model increases the likelihood of topic words appearing in the generated text to bias the overall distribution. The proposed TC-K2T can produce more informative and controllable senescence, outperforming state-of-the-art models, according to empirical research on automatic evaluation metrics and human annotations.

KEYWORDS

artificial intelligence techniques, artificial neural networks, deep learning

1 | INTRODUCTION

A keyword-to-text generation (K2T) problem seeks to create sentence-level texts that look like humans with only a few given keywords. Numerous application scenarios, including the generation of stories, reports, dialogue responses, second language, and other uses, have relied heavily on it [17, 25, 26, 29]. A great deal of interest has been drawn to K2T because of its enormous potential for practical use and scientific research. Despite this, two problems remain to be solved in K2T: 1) the neglect of controllable text generation from unordered keywords and 2) the underutilisation of topic-aware information.

An appropriately executed K2T generator should be able to generate a variety of vivid and varied sentences when keywords are used. However, existing work tends to produce generic and uncontrollable texts [24, 34, 39]. They ignore the information produced by text that is subject to topic control, which is one of the reasons. Our ability to produce text that is far more varied and fascinating is enhanced by modelling and controlling the subject matter that can be controlled by the generated text. As shown in Figure 1, given the keywords "Basketball", "Exercise", and "Game", the "without topic-controllable" models generate flat sentences. Meanwhile, the topic-controllable model generates controllable statements such as "The NBA is a game loved by basketball fans all over the world, and basketball is also a great exercise." when given the topic of "NBA", and generates phrases such as "The major outlet for basketball kit, Nike, has been promoting the exercise of basketball and has sponsored many basketball games." when given the topic of "Nike". Additionally, topic control is critical to the K2T generation process, which aims to generate a variety of sentences. The search space for the generation model multiplies exponentially when the topic polarity for each sentence is controlled as the number of words increases. For the task of K2T generation,

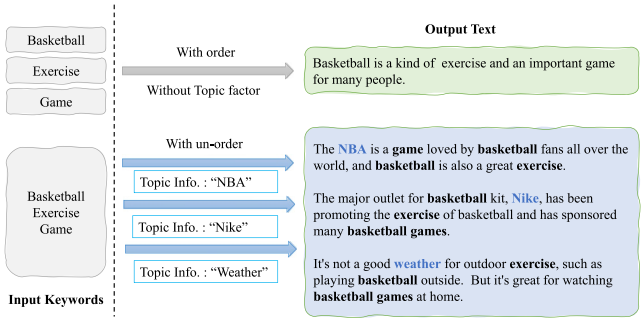

FIGURE 1 Examples of comparison between the generated text from ordered keywords with topic control and unordered keywords without topic control. We show the first three sentences for each generated text and denote topic words in blue and keywords in black bold. Sentences without topic factor are shown in a green text box.

therefore, the ability to manage topics is essential to enhance diversity at the discourse level.

The fact that we humans rely heavily on our common sense knowledge when asked to write sentences with some keywords and related topics should also be taken into account. Because of this, K2T generation relies heavily on the proper utilisation of knowledge. Early cutting-edge approaches based on fixed templates used a set of keywords and partial speech as input [22]. Nevertheless, they disregard the network structure of the knowledge base, which only makes reference to concepts in the knowledge network [4] and neglect to consider their correlations, as well as the topic-controllable information. Due to this restriction, conceptions become dissociated from each other. As an illustration, given two keyword corpus inputs, delight, antonym, sadness, and delight, part of, emotion about the topic word "delight", simply use the neighbouring concepts sadness and emotion as a complement to the input information. While "emotion", which is a hypernym to "delight", is a hypernym that can be learnt from their correlations (edges) in the knowledge network, their approach fails to recognise that "sadness" has the opposite meaning from "delight". Intelligently, the lack of information about the relationships between concepts in a knowledge network makes it difficult for a model to construct useful and informative texts.

This article explores a novel topic-controllable keywords-to-text generator with a topic information network decoder called TC-K2T that is based on a proposed conditional language encoder framework in order to address the aforementioned issues. As part of our model's encoder and decoder, we inject topic-controllable information to control text content from unordered keywords in order to control the subject from two perspectives: word-level and sentence-level. The label of each topic is provided by a topic classifier during the training process. Based on ConceptNet [30], a large-scale common sense knowledge base, the model recovers a topic knowledge network in order to fully utilise the information. Instead of preserving the network structure of the knowledge base [38], we provide a novel Topic Attention (TA) mechanism that is distinct from many existing methods. In order for future generations to benefit from structured, topic-controllable, connected data from networks, the TA conducts an in-depth review of knowledge networks. As a result, we employ adversarial training based on the multi-label discriminator to make the generated text more closely related to the topic-controllable information and to include all input keywords. Depending on how much the output covers the given keywords, the discriminator rewards the generator.

In conclusion, we make the following significant contributions:

- We propose a novel topic-controllable keywords-to-text generator using the conditional language framework that is capable of producing high-quality text and controlling the subject. According to our knowledge, we are the first to apply topic-controllable information to the task of keyword-to-text generation and demonstrate the potential of our model to generate diverse text by controlling the topic at the sentence level.
- We propose an innovative Topic Attention (TA) mechanism and use a topic knowledge network to enhance our decoder. TAs make the most of the structured, aggregated subject data from the subject knowledge network, and they are able to produce text that is more pertinent and informative.
- With the aid of extensive experiments, we validate that our model accurately controls the topic for text generation and outperforms cutting-edge methods in both automatic and human annotation.

2 | RELATED WORK

A growing portion of research is being done on text generation as people rely more and more on automatic text generation in their daily lives. A fundamental model for generating text is the Seq2Seq model, which is based on attention. Among the tasks for text generation that the attention-based Seq2Seq model is effective at are neural machine translation, abstract text summarisation, dialogue generation etc. In general, the Seq2Seq model has developed into one of the most well-known text generation frameworks.

RNNs are the foundation of the majority of Seq2Seq models, but recent research has led to frameworks based on CNNs and attention systems. In neural machine translation, the transformer has produced cutting-edge results and is rapidly becoming a popular framework for sequence-to-sequence learning as a result of its excellent performance and high efficiency. A transformer-based pre-trained language model called BERT is proposed to perform natural language processing tasks with the most sophisticated performance.

2.1 | Controllable text generation

A challenging task in natural language processing is the automatic generation of reliable, logical, and understandable text [27, 28]. For the first time, K. Uchimoto et al. [33] came up

with a framework for generating sentences based on n-grams and dependency trees. They created the framework solely for the Japanese people. For generating the context before and after a single keyword input in Chinese, a recurrent neural network RNN-based model [32] was recently employed.

Methods for managing style for tasks involving text generation have been the subject of some research. Artificial inputs that can be used to generate controlled text have received considerable attention in the text-to-text domain [14]. Another recently proposed approach is the controllable plug-n-play language model developed by Dathathri et al. [6]. Although their generator is able to generate fluent output based on the control specification, the generation process is still open-ended and may not adhere to any user-desired syntax. An alternative framework for variational auto-encoders (VAE) [11], which offers minimal control options, including sentiment, has been created. According to Ghosh et al. [10], a method can be employed to determine the degree of emotional content in generated sentences. Moreover, since the system relies heavily on actual textual data annotated with these categories and has a fixed set of emotion categories, it is unable to accept certain approaches. The linguistic properties of a text are controlled by a language model that is influenced by a particular style in a similar effort by Ficler and Goldberg [9]. As a result, there are several possible styles, including theme, sentiment, professional, and descriptive that they use in the movie review industry. Although the system lacks the ability to transform data, these styles may require only a small number of values to which the generated text ought to adhere. In the context of modern English texts, Jhamtani et al. [13] investigate an approach to applying the Shakespearean English style. In order to replace words by copying the style, the model employs an external dictionary of stylistic words; this might not always maintain the intended meaning.

2.2 | Topic-controllable generation

A wide variety of research initiatives on topic-controlled generation employ templates to control the direction of the sentences generated. Using templates provided in the form of "sparse" trees that are frequently used in a language, Iyyer et al. [12] propose a syntactically controlled paraphrase network (SCPN). By relying on well-formed sentences and the accompanying complete parse trees, the system is unable to transform the input into data (shown by keywords). The fact that the system can accept input templates is noteworthy because they are both syntactically rigid and difficult to interpret. Chen et al. [3] proposes an approach that uses a sentence as a syntactic example rather than requiring an external parser. Although this system can take up keywords in any order, it is not intended to accept data/keywords for input (different from our system). Recently, a method [35] inspired by the data-to-text generation dataset generated sentences given a structured record and a reference sentence. For fidelity to the structured material, manipulating the reference text (by rewriting, adding, or deleting portions of the text) is a different task. Our analysis reveals that keywords are not organised, even ordered, and may require morphological, syntactic, and numerical transformations (such as number, tense, and aspect change); as a result, it is not feasible to modify, add to, or delete portions of text. Similar to the aforementioned, Laha et al. [16] propose a modular system that converts input from structured data (tables) into a canonical form, develops straightforward sentences from canonical data, and ultimately combines sentences to produce a coherent and fluent paragraph statement. This approach, which involves table row representations as a collection of binary relationships (or triples), differs from ours in terms of the task size.

As far as we are aware, there are still numerous challenges in translating order-invariant keywords into natural language text without subject-aware information.

2.3 | Problem formulation

In this section, we first introduce some fundamental concepts that are necessary to understand our model. The notations used in this paper are summarised in Table 1.

Moreover, we formulate the problem of automatic controllable language generation. The objective of the task is to build a system that can generate topic-aware sentences automatically based on the input keywords. Given the keywords represented as an input sequence of words $k = [k_1, k_2, \ldots, k_N] \in K$, the objective of the system is to generate the topic-aware sentences $s = [s_1, s_2, \ldots, s_M] \in S$, a sequence of words describing the topic.

3 | FRAMEWORK ARCHITECTURE

For generating topic-aware text from keywords, the framework takes as input a set of $\mathcal{N}$ keywords denoted as $K = [k_1, k_2, \ldots, k_N]$ and aims to generate text with $\mathcal{M}$ sentences $[s_1, s_2, \ldots, s_M]$

TABLE 1 Notations.

Notation	Description
$\mathcal{N}$	The number of input keywords
K	Keywords array $K = [k_1, k_2, \ldots, k_N]$
S	Output sentences array $[s_1, s_2, \ldots, s_M]$
T	Topic sequence array $T = [T_1, T_2, \ldots, T_M]$
$S_{1:i-1}$	The previous sentences as context
$KL\ (q\|\|p)$	The KL annealing technique
$softmax\ (.)$	The softmax operation
C	Control code
Ss	The context S encoded as $[s_1, s_2, \ldots, s_{i-1}]$
$\mathcal{N}$	The prior network
Q	Query vector
W_α	The weight matrices

corresponding to keyword sets K. In addition, in this research, we provide a topic sequence $T = [T_1, T_2, \ldots, T_M]$, each of which corresponds to a specific sentence in text. Entities or virtualities can be used for each topic. In the sequence of input topics, each keyword influences each other, without a strong sequential relationship in this study, as depicted in Figure 2 through bidirectional arrows. Similarly, the sequence of input sentences is also encoded bidirectionally. Expanding on this, the lack of strong sequentiality in the influence between keywords within the topic input sequence means that the impact of one keyword on others is not solely determined by its position in the sequence. Instead, every keyword has the potential to affect and be affected by the other keywords, regardless of their order. When it comes to encoding the input sentences, bidirectional encoding allows for a comprehensive understanding of the context in which the sentences are presented. This means that not only the preceding sentences can influence the current sentence, but the subsequent sentences can also provide valuable information for understanding the current sentence.

The sentence-by-sentence process is used to create a continuous paragraph of text. After generating the first sentence s_1 based solely on the keyword sets K, the model continues to generate the following sentence using all the previously generated sentences and keyword sets until the entire text is finished. In this paper, the preceding phrases are represented as $S_{1:i-1}$.

The overall architecture is given in Figure 2, where $\oplus$ represents the vector concatenation operation. The $KL\ (q\|p)$ represents the Kullback–Leibler (KL) annealing technique [2], which is a commonly used approach in deep learning models, specifically in variational autoencoders. It introduces a gradual increase in the weight of the KL divergence term in the loss function during training. This technique addresses the issue of posterior collapse, where the posterior distribution collapses to a fixed point. By gradually increasing the weight of the KL term, the model is encouraged to capture both the global structure of the data and the fine-grained details, leading to improved convergence and overall performance.

Topic sequence T denotes subject control. The orange solid arrows represent the TA process at each decoding step. The text generated by the TC-K2T generator is fed into the topic-switching decision modules. The output grey blocks representing the given text are generated after a *softmax* (.) operation.

To incorporate the knowledge graph into the Conditional Variational Autoencoder (CVAE), we modify the loss function and introduce the Knowledge-Guided CVAE (kgCVAE) [42]. The kgCVAE loss aims to align the latent representations with the structured information provided by the knowledge graph. Specifically, we utilise a graph-matching objective that encourages the learnt latent space to encode relationships consistent with the knowledge graph. By optimising this loss during training, we can effectively leverage the knowledge graph to guide the generation and inference processes of the CVAE model. Based on the kgCVAE strategy consisting of an encoder and an enhanced topic knowledge network decoder), we have developed our TC-K2T generator.

Keywords, topic sequences, and context are encoded by the encoder and are viewed as conditional variables c. A latent variable is then calculated from c using a recognition network (during training) or a previous network (during inference). In our proposed method, a recognition network is used during the training phase to approximate the posterior distribution of the latent variables given the observed data. This recognition network is trained to map the data to a distribution in the latent space. In the subsequent inference phase, a previous network, which has been trained using the recognition network, is employed to generate samples or make predictions based on the learnt latent space representation.

A topic knowledge network and topic-related information are connected by the decoder to create texts. Through effective use of the topic knowledge network, TA is utilised at each decoding step to enhance input topic information.

As part of the training process, we take the following two steps: (1) Train the TC-K2T generator with the kgCVAE loss and (2) give a topic-controllable information discriminator to

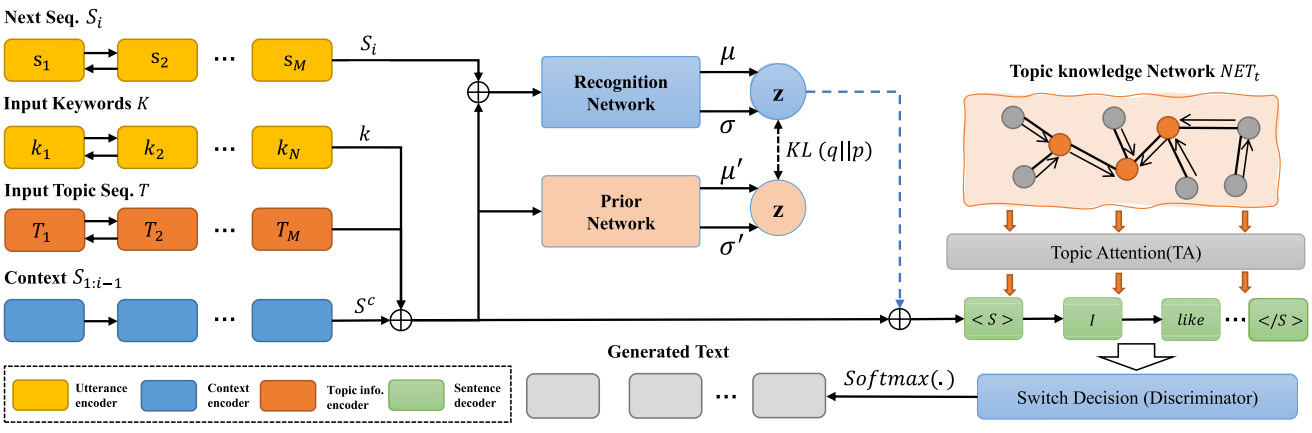

FIGURE 2 Our topic-controllable keywords-to-text generation framework. $\oplus$ denotes the vector concatenation operation. The part with solid lines and the blue dotted arrow is applied as inference, while the entire CVAE, except the blue dotted arrow part, is used in the training process. Orange solid arrows denote TA at each decoding step. Our text generator feeds to a topic-aware discriminator. The output grey blocks representing the text are generated after a *softmax* (.) operation.

evaluate the performance of the TC-K2T generator. In order to further enhance the TC-K2T generator's effectiveness, we utilise adversarial training to train both the generator and the discriminator occasionally.

4 | TOPIC-CONTROLLABLE KEYWORDS-TO-TEXT MODEL

4.1 | Encoder part

In a vector of the same size, the keyword encoder aims to capture contextual representations of each keyword. In addition, it should be ensured that the encoding process is insensitive to the order of the input keywords. Using the last hidden states of the forward and backward Gate Recurring Unit (GRU) [5] as our utterance encoder, we utilise a bidirectional GRU to produce input sets in a vector of a fixed size. We use the utterance encoder to encode the keyword sets K into $u^k = \left[\overrightarrow{h^k}, \overleftarrow{h^k}\right], h^k \in \mathbb{R}^d$ in which d is the vector dimension. The next sequence S_i is also encoded by utterance encoded as $u^i = \left[\overrightarrow{h^i}, \overleftarrow{h^i}\right]$, $h^i \in \mathbb{R}^d$. According to the context encoder, inspired by [42], we utilise a strategy of multi-layer encoding. For each sentence, firstly, in context $S_{1:i-1}$ is encoded by the utterance encoder to get a fixed-size vector. By doing so, the context $S_{1:i-1}$ is encoded as $s_{text} = [s_1, s_2, \ldots, s_{i-1}]$. Once this has been accomplished, a 1-layer forward GRU is used to encode sentence representations s_{text} into a final state vector $s^c \in \mathbb{R}$.

We then concatenate s^c, u^k and $e(t)$ (the embedding of topic information label), and define the conditional vector as $c = [e(t)|s^c, u^k]$. Since we assume that z follows isotropic Gaussian distribution, the recognition network $q_\phi(z|s_i, c)$ and the prior network $p_\theta(z|c)$ follow $\mathcal{N}(\mu, \sigma^2\mathbf{I})$ and $\mathcal{N}(\mu\prime, \sigma\prime^2\mathbf{I})$, respectively. I is the identity matrix and then we have

$$\begin{aligned}\left[\mu, \sigma^2\right] &= \mathrm{MLP}_{\mathrm{recognition}}\ (s_i, c)\\ \left[\mu\prime, \sigma\prime^2\right] &= \mathrm{MLP}_{\mathrm{prior}}\ (c).\end{aligned} \tag{1}$$

We then use the reparametrisation trick [42] to obtain samples of z either from $\mathcal{N}(z|\mu, \sigma^2\mathbf{I})$ predicted by the training recognition network or $\mathcal{N}(z|\mu\prime, \sigma\prime^2\mathbf{I})$ predicted by the testing prior network.

4.2 | Topic-controllable decoder

In general, Seq2Seq models can produce sentences that are generic and meaningless. An enhanced topic knowledge network decoder is proposed to produce a more meaningful text. The decoder is based on a 1-layer GRU network with an initial state $s_0 = W_d\,[z, c, e(t)] + b_d$. W_d and b_d are trainable decoder parameters, and $e(t)$ is the embedding topic as mentioned above. As shown in Figure 2, we built the decoder with a topic knowledge network to incorporate commonsense knowledge from ConceptNet [1] [31]. A semantic network called ConceptNet is intended to assist computers in understanding the meanings of words that people utilise. As triples of the start, connection label, and end nodes, this type of network is represented. The relationship exists between the end node and the start node. We use word vectors to represent start and end concepts and learn a trainable vector v^{rel} for the relation, which is randomly initialised. Our approach consists of learning trainable vectors for relations that are randomly initialised and using word vectors to represent start and end concepts. Using each word in the keyword sets as a query, ConceptNet is used to locate a subnetwork that forms the topic knowledge network. After that, we use the Topic Attention (TA) mechanism to read from the topic knowledge network at each generation stage.

It is essential for the success of our work that external expertise is used properly, as already stated. TA takes as input the retrieved topic knowledge network and query vector q to produce a network vector NET_t. We set $q = [d_{t-1}, c, z]$, where d_{t-1} represents the hidden state of the decoder for step $t-1$, c is the conditional vector and sample z from the recognition network.

Our algorithm calculates the correlation score between each of the triples in the network and q during decoding at each stage t. After that, the weighted sum of all the neighbouring concepts to the topic terms is calculated using the correlation score to create the final network vector NET_t.

According to reports, neighbouring things are those that are directly connected to topic terms. We denote the embedding of nth neighbouring concept as o_n, then NET_t can be defined as follows:

$$NET_t = \sum_{n=1}^{N} \alpha_n \boldsymbol{o}_n \tag{2}$$

In order to capture important information, we focus on the decoding process. The attention weights on the query are computed by the following equation:

$$\frac{\exp\left(f_n\right)}{\sum_{j=1}^{N} \exp\left(f_j\right)} \tag{3}$$

where

$$f_\alpha = \left(\mathbf{W}_\alpha \mathbf{q}\right)^T \tanh\left(\mathbf{W}_\alpha \mathbf{v}_n^{rel} + o_n\right) \tag{4}$$

A weight matrix for queries, relationships, start entities, and end entities is represented by W_α. Additionally, adjacent concepts, which are the start/end ideas in their triples, fall under the category o_n. The correlation between the query q and the

[1] https://conceptnet.io

neighbouring concept o_n is represented by the matching score f_α. It can measure the topic relationship between the ith word in the source and the jth target word to be predicted. In essence, it makes up a network vector NET_t by combining adjacent concepts of topic words. Be aware that different weight matrices are utilised to distinguish between the neighbouring concepts in various positions (in start or in end). This distinction is necessary, for instance, in the light of two triples of knowledge (Opera House, part of, Sydney) and (Sydney, part of, Australia). The concepts of Opera House and Australia have different meanings for Sydney despite the fact that they are both adjacent concepts to Sydney with the same connection component. We need to model this difference in the weight matrices set W_α.

In order to calculate the final chance of generating a word, the following steps must be taken:

$$\mathcal{P}_t = \text{softmax}(\mathbf{W}_o[d_t; e(t); NET_t] + b_o), \tag{5}$$

where d_t is the decoder state at t step and $\mathbf{W}_o \in \mathbb{R}^{d_{all} \times |V|}$, $b_0 \in \mathbb{R}^{|V|}$ are trainable decoder parameters, d_{all} is the dimension of $[d_t; e(t); NET_t]$ and $|V|$ is the vocabulary size.

A strong correlation needs to be established between the generated text and keywords and topic terms. We use a soft switcher to figure out if a word should be generated as the target word by using $\lambda_j \in [0, 1]$:

$$\lambda_j = sigmoid(\mathbf{W}_\lambda[e(t)]) \tag{6}$$

with W_λ being a learnable parameter. This section also contains information that can be controlled by topics $e(t)$ to guide the switch selection. Further, the sigmoid probability distribution over $(m + 1)$ classes [36]. According to the $(m + 1)^{th}$ index, the probability that the sample is the generated text is represented by the score. The likelihood of it being a real text with the *jth* topic is represented by the score on the *jth* index.

4.3 | Model training

Throughout this section, we discuss the two-step training method. The first one is similar to a conventional kgCVAE model. The loss of our TC-K2T generator $-logp\,(Y|c)$ can be defined as follows:

$$\begin{aligned} -\mathcal{L}(\theta; \phi; c; Y)_{kgcvae} &= \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{decoder}} \\ &= \text{KL}\Big(q_\phi(z|Y,c) \| p_\theta(z|c)\Big) \\ &- \mathbb{E}_{q_\phi(z|Y,c)}\big(\log p_{\text{D}}(Y|z,c)\big) \end{aligned} \tag{7}$$

This parameter list includes θ and ϕ for the recognition network and the prior network, respectively. Intuitively, $\mathcal{L}_{\text{decoder}}$ maximises the sentence generation probability after sampling from the recognition network, while $\mathcal{L}_{\text{KL}}$ minimises the distance between the prior and the recognition network. Our adversarial training between the generator and the topic label discriminator described above begins after training the TC-K2T generator with Equation 7, inspired by SeqGan [40]. Additional information is provided to the reader using the SeqGan approach due to page restrictions. Besides, we use the annealing trick and BOW-loss equation to alleviate the vanishing latent variable problem in VAE training.

5 | EXPERIMENTS

Throughout this section, we discuss the dataset, evaluation metrics, all baselines, and settings in more detail.

5.1 | Datasets

We conducted experiments on the Quora corpus [2]. Experiments were done on the datasets, which are between 30 and 120 words in length. In light of the frequency of each keyword, we choose words from the NOUN, VERB, ADJECTIVE, and ADVERB categories as input keywords and remove uncommon keywords. There are 30,000 and 3000 test sets for training and testing, respectively. As the validation setting, we utilised 12% of the training samples to tune the hyperparameters.

Our method of introducing topic labels involved manually annotating items with 100 categories, such as sports, beauty, and so on, using 3000 sentences. To fine-tune our manually labelled training set, we utilise the topic model proposed by Zandie et al. [41], which achieves an accuracy of 0.87 on the test set. By using an automatic topic extractor during training, the target topic label s is calculated. The direction of each text that is generated is controlled by user input of any topic labels throughout inference.

5.2 | Settings

In order to implement topic embeddings, we utilise 120-dim pre-trained word embeddings with 32 dimensions. The vocabulary size is 50,000, and the batch size is 64. Selecting the hyperparameter values uses a manually tuned procedure, and the criteria chosen is BLEU. We employ GRUs with a hidden size of 512 and a hidden size of 300 for both encoders and decoders. We develop the model with the Tensorflow framework [3]. With a total parameter count of 72 MiB, our model parameters were randomly selected over a uniform distribution [0.1, 0.1]. We pre-train our model for 65 epochs with the Maximum Likelihood estimation model [41] and adversarial training [7] for 20 epochs. Our model is pre-trained with the Maximum Likelihood estimation model for 65 epochs and with adversarial training for 20 epochs. The average runtime for our model is 35 h on an Intel(R) i7 CPU @ 2.50 GHz, 512 GB

[2] https://www.kaggle.com/competitions/quora-question-pairs/data
[3] https://github.com/tensorflow/tensorflow

RAM and 2 NVIDIA 1060Ti-16 GB GPUs. The optimiser is Adam [15] with 10^{-3} learning rate for pre-training and 10^{-5} for adversarial training. In addition, to prevent overfitting [21] with the dropout rate of 0.2 and gradients to the maximum norm of 10, we use dropout on the output layer. The average length of the generated text is 56.2, and greedy search is used in our model's decoding strategy.

5.3 | Evaluation metrics

Both human annotations and automatic evaluation are employed by us in order to thoroughly assess the generated text.

BLEU [23]: By measuring word overlap between ground truth and generated sentences, the BLEU score is frequently used in machine translation, conversation, and other text generation tasks.

Distinct-1 & distinct-2: Several distinct bigrams and unigrams were taken into account in the responses generated. Furthermore, we split the numbers by the total number of unigrams and largerams using [18]. We define metrics as distinct-1 and distinct-2, respectively, for both numbers and ratios. Using both metrics, one can assess how informative and varied the text produced is. In addition, high numbers indicate that the produced text is lengthy, and high numbers and high ratios indicate that there is a lot of content in it.

Consistency [38]: Using all the keystrokes entered, a suitable text should surround a particular topic closely. For the purpose of evaluating the topic consistency of the output, we use a multi-label classifier pretrain. Higher scores on "Consistency" indicate that the generated texts are more closely related to the given topics. Taking into account the input topics T, the topic consistency of the generated text $\hat{\boldsymbol{y}}$ is defined as follows:

$$\text{Consistency}(\hat{\boldsymbol{y}}|\boldsymbol{x}) = \varphi(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{8}$$

where φ is the Jaccard similarity function and $\hat{\boldsymbol{x}}$ is the topic predicted by a pre-trained multi-label classifier.

Perplexity: Following [20], we employ perplexity as an evaluation metric. Perplexity is defined by Eq. (9). Higher generation performance is associated with a lower perplexity rating. The purpose of this work is to determine when training will end using PPL(D). For five validation scenarios, if the perplexity stops decreasing and the difference is less than 2.0, we think the algorithm has reached an agreement and the training is terminated. In the test data, we use PPL(T) to evaluate the ability of various models to be produced.

$$PPL = \exp\left\{-\frac{1}{N}\sum_{i=1}^{N}\log(p(\mathbf{Y_i}))\right\} \tag{9}$$

Novelty [38]: Our analysis of novelty took into account the difference between the output and text with similar subjects in the training corpus. Increased "Novelty" scores indicate that the text in the output corpus is more distinct from that in the training corpus.

Precision, Recall and F1: For the purpose of determining the accuracy of theme control, these metrics are used. A valid result can be obtained if the topic labels on the generated sentences are consistent with the ground truth. We predict topic labels using the topic classifier.

Human annotation: For the purpose of evaluating the quality of the created text of various models, we also employ human annotators. Three annotators with extensive Quora knowledge are invited to participate in the evaluation. Random shuffling and pooling of text created by different models are performed for each annotator. Test messages are reviewed by annotators who evaluate the quality of the text according to the following criteria:

Level-4 (L4): The generated text shows the topic consistency and novelty. Meanwhile, the text represents not only the text diversity but also the natural and fluent degree.

Level-3 (L3): The output text has a clear topic-aware direction, and the sentence pattern is natural and smooth.

Level-2 (L2): The output text is just natural and informative.

Level-1 (L1): The text is difficult to understand and is either semantically irrelevant or disfluent.

5.4 | Baseline

Based on previous state-of-the-art text generation approaches, we make the following baseline measurements:

S2SA-MMI: With an attention mechanism in a standard Seq2Seq model, it performs at its best in Ref. [19]. We utilised the bidirectional-MMI decoder and, in accordance with the paper's suggestions, set the hyperparameter λ to 0.5.

TRANS: Keywords can only be entered without requiring any controllable operations [22].

NMT [1]: The purpose of this paper is to describe a neural machine translation that uses an encoder-decoder framework based on LSTM and only accepts keywords as input without any topic control mechanisms.

CONCAT [22]: Words and templates are concatenated as input by the transformer-based framework.

TAV [8]: All topic embeddings' average topic semantics are used to create each word using an LSTM. The semantic correlation between each topic word and the output of the generator is modelled using an attention mechanism that extends LSTM.

CTEG [38]: To improve the generation process, a combination of common sense knowledge and adversarial training was suggested. This work achieves state-of-the-art performance on the topic-to-essay generation task.

5.5 | Quantitative performance comparison

As shown in Table 2, we list automatic and human evaluation results. We provide three different versions of our model for a

TABLE 2 Automatic and human annotations result. In human annotation, there are four levels (L4 - L1) to quantify: topic consistency, novelty, text diversity, fluency and informative. The best performance is underlined.

	Automatic evaluation							Human annotation			
Methods	BLEU	Dist-1	Dist-2	Consis.	PPL(T)	PPL(D)	Novelty	L4	L3	L2	L1
S2SA-MMI [37]	6.07	4.81	21.64	9.20	147.04	143.11	67.98	0.13	0.23	0.38	0.26
TRANS [22]	6.32	5.01	22.03	26.51	134.23	144.47	71.23	0.19	0.34	0.21	0.26
CONCAT [22]	7.01	5.12	22.54	35.54	141.45	156.91	72.45	0.27	0.23	0.25	0.25
TAV [8]	6.52	5.32	22.43	16.57	131.67	148.52	69.45	0.23	0.18	0.29	0.30
NMT [1]	7.12	5.31	22.67	32.67	128.63	149.34	72.34	0.28	0.21	0.24	0.27
CTEG [38]	9.72	5.92	23.07	39.32	127.27	142.37	73.39	0.31	0.18	0.31	0.20
TC-K2T (w/o-Topic)	10.01	5.64	23.21	44.12	119.55	140.45	78.26	0.36	0.22	0.21	0.21
TC-K2T (R-topic)	9.98	5.86	23.11	42.01	122.81	133.81	80.12	0.42	0.27	0.17	0.14
TC-K2T (Pro-topic)	12.01	6.01	23.08	42.63	123.14	134.63	78.87	0.45	0.24	0.19	0.12

comprehensive comparison. (a) "TC-K2T (w/o-Topic)" means that we do not attach any topic information to the model. (b) "TC-K2T (R-Topic)" means that we randomly set the topic information for each generated text. (c) "TC-K2T (Pro-Topic)" represents how we set the high-frequency topic information used for sentence generation. The results in Table 2 show the following conclusions:

- Whether it is with hot topics, random topics, or without topics, all variants of our models outperform the baselines in all evaluation metrics, demonstrating the proposed TC-K2T model's capacity to generate better texts than baseline models.
- The superiority of our model architecture can be seen in the comparison of TC-K2Ts (w/o-Topic) and baselines. Based on human annotation results, TC-K2T (Pro-Topic) performs best on level-4 metrics. Most significantly, there has been an improvement in text diversity. We achieve this improvement with our kgCVAE architecture because sampling a continuous latent variable serves as our sentence representation. Compared to baselines, this sampling procedure introduces more randomness.
- The "L4" texts exhibit a 0.29 increase in TC-K2T (R-Theme) compared to S2SA-MMI, while the "L1" texts display a 0.12 decrease. Compared to TC-K2T (w/o-Topic), TC-K2T (R-Topic) performs better, demonstrating that topic content contributes more to text quality than bias probability in generation.
- Thermal topic information is used by TC-K2T (Pro-Theme), which performs well in BLEU with a score of 12.01. In other metrics, TC-K2T (Pro-Theme) does not significantly outperform other TC-K2T models, according to our analysis. These results demonstrate that our suggested model is more appropriate for the text set because of the hot topic information of the target texts, although there is no obvious improvement for other important indicators such as Distinct, Consistency, and Perplexity.
- When the topic information is removed, we find it intriguing that TC-K2T (w/o-Topic) achieves the best topic-consistency score. However, the effect of this interference is trivial because we believe that topic labels may somehow interfere with the subject information in the latent variable. For automated evaluation, we find that the topic consistency for TC-K2T (w/o-Topic) and TC-K2T (Pro-Topic) drops by only 1.49 (44.12 vs. 42.63), which is completely acceptable for a model that can handle subjects.

5.6 | Text quality analysis

Both ablated versions of our main model are trained to better comprehend how each component of our model contributes to the task: 1) removing adversarial training—"w/o AT", 2) removing TA - "w/o TA". Moreover, we employ a memory network [30] in the "w/o TA" experiment that incorporates ConceptNet concepts but does not consider their correlation. All models use frequency topic words. According to the findings in Table 3, the human annotation and BLEU scores of the ablation study are presented. A score for the generated text is obtained using [1:5] using four metrics (Novelty, Fluency, Topic Consistency, and Text Diversity) in order to assess the quality of the text. In order to offer annotators a reference, we use the TF-IDF features of topic words to find the 20 training samples that are most similar for "novelty".

With the exception of the topic attention layer, we find that the full model and "w/o TA" both have lower model performance in all metrics. For instance, topic consistency dropped by 0.31, demonstrating that concepts with stronger connections to subject words receive greater attention during generation when the relationship between those concepts and the topic words is explicitly learnt. TA is an expansion of the information contained in the external knowledge network, resulting in a drop of 0.08 in novelty. As a result, the text output is more novel and informative. The TA provides our model with the benefit of selecting a concept that is more appropriate in the topic knowledge network in the current context, which results in a 0.42 decrease in fluency. In

addition, the BLEU decreases to 1.69, demonstrating TA's contribution to our model's ability to better fit the dataset by modelling the connections between topic words and nearby concepts.

We find that adversarial training can improve BLEU, topic consistency, and fluency by comparing the complete model and the "w/o AT". As a result, the discriminative signal increases the topic consistency and authenticity of the texts that are generated.

5.7 | Topic control analysis

Our focus in this section is on whether the model properly regulates the topic and how each component influences our topic control performance. Our model is trained in three ablated versions: 1) without topic information in the encoder, 2) without subject information in the decoder, and 3) without TA. We randomly sampled 120 texts in our test set with 510 sentences. Instead of using frequent topic words, all topic inputs are randomly given in this section. Predicting the topic is relatively straightforward because there are times when these types of terminology can be directly related to context consistency. To generate sentences based on arbitrary information about the topic, we employ a more difficult experimental setting.

According to Table 4, removing the topic embedding from the encoder or decoder has the greatest impact on control performance, and the topic embedding in the encoder is the most significant since removing it results in the greatest decline. It demonstrates that learning correlations between concepts in the topic knowledge network enhances the model's ability to control topics even though TA does not directly impose topic information. For example, when giving information about a "sports" topic, concepts related to the relationship "basketball games" are more likely to gain attention because concepts with the relationship "basketball games" have a certain probability of matching "sports" meaning.

TABLE 3 Text quality analysis results, "w/o AT" represents without adversarial training. "w/o TGA" represents without TA. T-Con. (topic-consistency), Nov. (novelty), T-div. (text-diversity) and Flu. (Fluency) represents different text qualities. The full model shows TC-K2T (Pro-Topic) in this table.

Methods	BLEU	T-Con.	Nov	T-div.	Flu.
Full model	**12.01**	**3.92**	**3.26**	**4.01**	**3.81**
w/o TA	10.32	3.61	3.18	3.89	3.39
w/o AT	9.72	3.31	3.51	3.92	3.54

TABLE 4 Topic control analysis. "w/o En-topic" represents removing the topic embedding in the encoder process, and "w/o De-topic" represents removing from the decoder. The full model represents TC-K2T (R-Topic) in this table.

Methods	Pre.	Recall	F1
Full model	**0.71**	**0.69**	**0.68**
w/o En-topic	0.53	0.52	0.53
w/o De-topic	0.56	0.63	0.62
w/o TA	0.63	0.65	0.64

5.8 | Case study

In this section, we present a case study of the texts we have actually created. Table 5 presents an instance of our output text with a random topic sequence. Keywords are shown in blue, and topic-controllable words are shown in red. As we learn, the output text is closely related to the topic words in addition to covering all the input keywords. The TA assists us in developing our model, which makes full use of common sense knowledge. For example, "fibre and antioxidants" and "reducing the risk of cancer" are correlation concepts related to the topic words "Health".

6 | CONCLUSIONS AND FUTURE WORK

This paper proposes a novel topic-controllable text generator with an enhanced decoder for topic knowledge networks called the TC-K2T model, which is the first attempt at the challenging task of keyword-to-text generation. On a public dataset, a number of experiments are conducted to evaluate the TC-K2T model's performance and demonstrate that it is superior to cutting-edge models.

Considering the recent success of multimedia-sharing platforms, the content posted on these online social media websites contains a wealth of multimedia information (e.g., text and images). A fascinating future course would involve exploiting these multi-modality features. In addition, extending our model to address multimodal generation by introducing topic interaction and label information can be considered a new research line in this field.

TABLE 5 Given keywords "Cabbage", "Vegetable", "Diet", and "Option", and set a topic word "Health". We generated a text according to the topic with keywords.

Input keywords: Cabbage, option, Vegetable, diet
Input topics: Health
Output text:
Cabbage is a leafy green vegetable that is often overlooked, but it is actually quite healthy. Cabbage is a good source of vitamins C And K, as well as fibre and antioxidants. Additionally, Cabbage has been shown to have a number of health benefits, including reducing the risk of cancer, improving heart health, and aiding in digestion. So if you are looking for a healthy vegetable to add to your diet, Cabbage is a great option.

ACKNOWLEDGEMENT

Open access publishing facilitated by University of Technology Sydney, as part of the Wiley - University of Technology Sydney agreement via the Council of Australian University Librarians.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Li He https://orcid.org/0000-0002-4637-8733
Xianzhi Wang https://orcid.org/0000-0001-9582-3445

REFERENCES

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate (2014). arXiv preprint arXiv:1409.0473
2. Bowman, S.R, et al.: Generating sentences from a continuous space. (2015) arXiv preprint arXiv:1511.06349
3. Chen, M., et al.: Controllable Paraphrase Generation with a Syntactic Exemplar (2019). arXiv preprint arXiv:1906.00565
4. Chen, Q., et al.: Towards knowledge-based personalized product description generation in e-commerce. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3040–3050 (2019)
5. Chung, J., et al.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (2014). arXiv preprint arXiv:1412.3555
6. Dathathri, S., et al.: Plug and Play Language Models: A Simple Approach to Controlled Text Generation (2019). arXiv preprint arXiv:1912.02164
7. Ding, J., et al.: Reinforced Negative sampling for Recommendation with Exposure data. In: IJCAI, pp. 2230–2236 (2019)
8. Feng, X., et al.: Topic-to-essay generation with neural networks. IJCAI, 4078–4084 (2018)
9. Ficler, J. and Goldberg, Y.: Controlling linguistic style aspects in neural language generation. (2017) arXiv preprint arXiv:1707.02633
10. Ghosh, S., et al.: Affect-lm: a neural language model for customizable affective text generation. (2017) arXiv preprint arXiv:1704.06851
11. Hu, Z., et al.: Toward controlled generation of text. In: International Conference on Machine Learning, pp. 1587–1596. PMLR (2017)
12. Iyyer, M., et al.: Adversarial Example Generation with Syntactically Controlled Paraphrase Networks (2018). arXiv preprint arXiv: 1804.06059
13. Jhamtani, H, et al.: Shakespearizing modern language using copy-enriched sequence-to-sequence models. (2017) arXiv preprint arXiv: 1707.01161
14. Kabbara, J., Cheung, J.C.K.: Stylistic transfer in natural language generation systems using recurrent neural networks. Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods, 43–47 (2016)
15. Kingma, D.P. and Jimmy, B: Adam: a method for stochastic optimization. (2014) arXiv preprint arXiv:1412.6980 (2014)
16. Laha, A., et al.: Scalable micro-planned generation of discourse from structured data. Comput. Ling. 45(4), 737–763 (2020). https://doi.org/10.1162/coli_a_00363
17. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. (2019) arXiv preprint arXiv:1910.13461
18. Li, J., et al.: A diversity-promoting objective function for neural conversation models. Proc. of NAACL-HLT (2016)
19. Li, J., et al.: Deep reinforcement learning for dialogue generation. (2016) arXiv preprint arXiv:1606.01541
20. Xiang, L., et al.: Diffusion-LM improves controllable text generation. (2022) arXiv preprint arXiv:2205.14217
21. Luo, Y., et al.: Few-Shot Table-To-Text Generation with Prefix-Controlled Generator (2022). arXiv preprint arXiv:2208.10709
22. Mishra, A., et al.: Template controllable keywords-to-text generation. (2020) arXiv preprint arXiv:2011.03722
23. Kishore, P., et al.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02), pp. 311–318. Association for Computational Linguistics, USA (2002)
24. Shang, L., Lu, Z., and Li, H: Neural responding machine for short-text conversation. (2015) arXiv preprint arXiv:1503.02364
25. Shi, K., et al.: Automatic generation of meteorological briefing by event knowledge guided summarization model. Knowl. Base Syst. 192(2020), 105379 (2020). https://doi.org/10.1016/j.knosys.2019.105379
26. Shi, K., et al.: Multiple knowledge-enhanced meteorological social briefing generation. IEEE Transactions on Computational Social Systems, 1–12 (2023). https://doi.org/10.1109/tcss.2023.3298252
27. Shi, K., et al.: AMR-TST: abstract meaning representation-based text style transfer. In: Findings of the Association for Computational Linguistics (2023). ACL 2023. 4231–4243
28. Shi, K., et al.: LLaMA-E: empowering E-commerce authoring with multi-aspect instruction following. (2023) arXiv preprint arXiv:2308.04913 (2023)
29. Shi, K., et al.: EKGTF: a knowledge-enhanced model for optimizing social network-based meteorological briefings. Inf. Process. Manag. 58(4), 102564 (2021). https://doi.org/10.1016/j.ipm.2021.102564
30. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: Thirty-first AAAI Conference on Artificial Intelligence (2017)
31. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: an open multilingual graph of general knowledge. In: National Conference on Artificial Intelligence (2017)
32. Surya, S., et al.: Unsupervised neural text simplification. (2018) arXiv preprint arXiv:1810.07931
33. Uchimoto, K., Sekine, S., Isahara, H.: Text generation from keywords. In: COLING 2002: The 19th International Conference on Computational Linguistics (2002)
34. Vinyals, O. and Le, Q.: A neural conversational model. (2015) arXiv preprint arXiv:1506.05869
35. Wang, W., et al.: Toward Unsupervised Text Content Manipulation, pp. 91 (2019). arXiv preprint arXiv:1901.09501
36. Wiseman, S., Shieber, S.M., Rush, A.M.: Challenges in Data-To-Document Generation (2017). arXiv:1707.08052 http://arxiv.org/abs/1707.08052
37. Xing, C., et al.: Topic aware neural response generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
38. Yang, P., et al.: Enhancing topic-to-essay generation with external commonsense knowledge. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2002–2012 (2019)
39. Yao, L., et al.: Plan-and-write: towards better automatic storytelling. Proc. AAAI Conf. Artif. Intell. 33(01), 7378–7385 (2019). https://doi.org/10.1609/aaai.v33i01.33017378
40. Yu, L., et al.: Seqgan: sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
41. Zandie, R., Mohammad, H., Mahoor: Topical Language Generation Using Transformers (2021). http://arxiv.org/abs/2103.06434
42. Zhao, T., Zhao, R., Eskenazi, M.: Learning Discourse-Level Diversity for Neural Dialog Models Using Conditional Variational Autoencoders (2017). arXiv preprint arXiv:1703.10960

How to cite this article: He, L., et al.: A topic-controllable keywords-to-text generator with knowledge base network. CAAI Trans. Intell. Technol. 9(3), 585–594 (2024). https://doi.org/10.1049/cit2.12280