

# Conditional Text Generation for Harmonious Human-Machine Interaction

BIN GUO\*, Northwestern Polytechnical University, P.R.China  
 HAO WANG, Northwestern Polytechnical University, P.R.China  
 YASAN DING, Northwestern Polytechnical University, P.R.China  
 WEI WU, Microsoft corporation  
 SHAOYANG HAO, Northwestern Polytechnical University, P.R.China  
 YUEQI SUN, Northwestern Polytechnical University, P.R.China  
 ZHIWEN YU, Northwestern Polytechnical University, P.R.China

In recent years, with the development of deep learning, text generation technology has undergone great changes and provided many kinds of services for human beings, such as restaurant reservation and daily communication. The automatically generated text is becoming more and more fluent so researchers begin to consider more anthropomorphic text generation technology, that is the conditional text generation, including emotional text generation, personalized text generation, and so on. Conditional Text Generation (CTG) has thus become a research hotspot. As a promising research field, we find that many efforts have been paid to exploring it. Therefore, we aim to give a comprehensive review of the new research trends of CTG. We first summary several key techniques and illustrate the technical evolution route in the field of neural text generation, based on the concept model of CTG. We further make an investigation of existing CTG fields and propose several general learning models for CTG. Finally, we discuss the open issues and promising research directions of CTG.

CCS Concepts: • **Information systems** → **Social networks**; • **Human-centered computing** → *Collaborative and social computing*.

Additional Key Words and Phrases: Human-computer interaction, conditional text generation, deep learning, dialog systems, personalization

## ACM Reference Format:

Bin Guo, Hao Wang, Yasan Ding, Wei Wu, Shaoyang Hao, Yueqi Sun, and Zhiwen Yu. 2020. Conditional Text Generation for Harmonious Human-Machine Interaction. *ACM Trans. Intell. Syst. Technol.* 1, 1 (December 2020), 51 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

---

Authors' addresses: Bin Guo, [guob@nwpu.edu.cn](mailto:guob@nwpu.edu.cn)(Corresponding-author), Northwestern Polytechnical University, Xi'an, P.R.China; Hao Wang, Northwestern Polytechnical University, Xi'an, P.R.China; Yasan Ding, Northwestern Polytechnical University, Xi'an, P.R.China; Wei Wu, [wuwei@microsoft.com](mailto:wuwei@microsoft.com), Microsoft corporation; Shaoyang Hao, Northwestern Polytechnical University, Xi'an, P.R.China; Yueqi Sun, Northwestern Polytechnical University, Xi'an, P.R.China; Zhiwen Yu, Northwestern Polytechnical University, Xi'an, P.R.China.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org/permissions).

© 2020 Association for Computing Machinery.

2157-6904/2020/12-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

## 1 INTRODUCTION

Jorge Luis Borges<sup>1</sup> once described a magic Library, named “The Library of Babel”<sup>2</sup>, where everyone could find any book he wanted. The readers cannot help but wonder who wrote these books. Are they all written by human writers? Absolutely, the answer is no. This library seems unlikely to exist, however, the development of text generation technology in recent years has made it possible. For instance, Philip M. Parker, having written and sold more than 100,000 books on Amazon, utilizes computer programs to collect massive publicly information on the Internet for automatic compilation into books<sup>3</sup>. The above scene of Parker belongs to the *text-to-text* generation [44], which takes existing textual materials as input and automatically generates the new text.

Text-to-text generation is a typical subfield of *text generation* [95], which uses diverse types of information to enable computers to learn to express like human, including image, text and so on. According to different data sources, text generation can be divided into *data-to-text*, *text-to-text*, and *image-to-text* generation. News generation is a typical application of data-to-text generation. For example, there was an earthquake in California on March 17, 2014, and the Los Angeles Times firstly provided detailed information about the time, location and magnitude of the quake. Actually that news article was automatically generated by a ‘robot reporter’, which converted the incoming registered seismic data into text by filling slots in predefined templates [101]. The data-to-text generation technology fills the established template with structured data and generates the output text containing all key components, which has exerted considerable influence in the field of news media.

The application of text generation from text to text includes machine translation [22], dialogue system [115], text summarization [111], reading comprehension [51], etc. By understanding the original text and obtaining its semantic representation, natural language text is generated for communicating, summarizing or refining. Besides, The application fields from image to text generation include image captioning [91], visual question answering [2], etc. By processing image information, the contents contained in the image can be understood to generate corresponding natural language descriptions and answers.

Deep learning contributes to the most recent advances in the text generation field. Specifically, with the help of the recurrent neural networks (RNN) [36], attention mechanism, generative adversarial networks (GAN) [47], reinforcement learning (RL), Variational Autoencoder (VAE) [63] and Transformer [126], the generated text becomes more coherent, logical and emotionally harmonious, which is more suitable for offering assistance in every aspect of people’s lives. For example, the dialogue systems, such as Microsoft Xiaolce<sup>4</sup>, Cortana<sup>5</sup> and Apple Siri<sup>6</sup>, can not only chat with us, but also assist us to operate electronic devices. News-writing-robots have provided creative assistance for journalists, and the machine translation technology has effectively met our needs of translation.

Advancements in universal text generation technology prompt researchers to explore more anthropopathic text generation methods, such as context-based text generation [56], personalized text generation [86], topic-aware text generation [133], emotional text generation [66], knowledge-enhanced text generation [147] and visual text generation [26]. Obviously, applying additional information in text generation may make the generated text more personified and facilitate harmonious human-machine interaction. However, new challenges are raised, summarized as follows.

<sup>1</sup>[https://www.goodreads.com/author/show/500.Jorge\\_Luis\\_Borges](https://www.goodreads.com/author/show/500.Jorge_Luis_Borges)

<sup>2</sup><http://www.paulrittman.com/JLBRLibraryofBabel.pdf>

<sup>3</sup><https://www.kurzweilai.net/he-wrote-200-000-books-but-computers-did-some-of-the-work>

<sup>4</sup><http://www.msxiaobing.com/>

<sup>5</sup><http://www.msxiaona.cn/>

<sup>6</sup><https://www.apple.com/siri/>

- How to efficiently integrate the additional conditional information with traditional model structures is a big challenge.
- Due to the scarcity of text datasets with specific conditions, training the conditional text generation models become more difficult.
- There is no reasonable evaluation metrics of the conditional text generation, making it difficult to quantify the performance of models.

This paper aims to give an in-depth survey of the development of neural text generation models. Specifically, we mainly focus on various studies on *conditional text generation* (CTG), such as context-based text generation, topic-aware text generation, and knowledge-enhanced text generation. Compared with general text generation, the conditional text generation is more in line with the needs of precision and friendly services.

To sum up, we summarize the contributions of our work as follows.

- Based on a brief review of current text generation techniques, we characterize the concept model of CTG and present the major human-centric services.
- We make an investigation of several different CTG fields, including context-based text generation, personalized text generation, topic-aware text generation, emotional text generation, knowledge-enhanced text generation, visual text generation, multi-conditional text generation and pre-trained language model-based text generation. Besides, the evaluation methods and conditional datasets are also discussed. Based on the summary of existing researches, we propose several general learning models for CTG.
- We further discuss some promising research directions of CTG, including the consideration of different types of contexts, the multi-modal data translation, lifelong learning, and so on.

The remainder of this paper is organized as follows. In Section 2, we give a brief review of key text generation techniques. In Section 3, we characterize the concept model of CTG. We then summarize the major researches of CTG in Section 4 and propose several general learning models for CTG in section 5, followed by the open issues and future research directions in Section 6. Finally, we conclude this paper in Section 7.

## 2 THE KEY TECHNIQUES OF NEURAL TEXT GENERATION

The past few decades have witnessed a huge leap forward in the text generation technology. Specifically, from the original rule-based and statistical methods to the end-to-end neural network-based methods, the overall quality of generated content is further improved. This section gives a brief review of key techniques of neural text generation.

DNN-based methods do not require manual feature extraction, and can automatically learn the continuous vector representations in three steps for the task-specific knowledge in different tasks, i.e. encoding, reasoning and decoding successively [43]. The inputs of neural network models will be firstly encoded into a vector space, where semantically related or similar concepts are close to each other. Afterwards the neural networks will reason in the vector space according to the hidden state and current input to produce the system response. Finally, the system response will be decoded to generate natural language text. Different neural network structures are usually adopted in the stages of encoding, reasoning and decoding, and all the parameters are optimized through back-propagation and gradient descent algorithms. The end-to-end learning mechanism promotes neural networks to fully mine the correlation in the data, and alleviates the requirement of characteristic engineering greatly.

The key techniques of neural text generation mainly include RNN, GAN, RL, VAE, and Transformer, which will be summarized in the following subsections. Natural language text is a typical kind of sequential data with specific relationships between contexts, and the natural sequential

structure of RNN is very suitable for modeling text data. Since RNN contain internal memory, which can remember previous inputs and the current input, it makes sequence modeling much easier. The output at the current time step depends not only on the instantaneous input, but also on outputs of previous time steps, which makes it highly capable of capturing contextual information and generating sentences that satisfy syntactic structures. However, the word-by-word sequential generation process of RNN cannot learn representations of full sentences, making it tend to generate inconsistent and uninformative text because of the absence of global features like topics or high-level syntactic features. With latent variables in continuous space, VAE can capture implicit language structure and utterance-level semantics (e.g., topics, syntactic properties), which are served as the global representations during the decoding process. By the sampling procedure of the latent variables, VAE is capable of producing more natural, meaningful, and diversified natural language texts.

The RNN-based text generation models trained by maximizing the log-likelihood objective function are prone to the problem of exposure bias, caused by the inconsistency of the sequence input during training and testing. Therefore, GAN, another powerful deep generative model that has become a huge success in computer vision, is introduced to text generation which uses adversarial training to replace the maximum likelihood training to simulate the real data distribution and generate higher-quality text. Through the adversarial training of the generator and discriminator, the generator of GAN gains the ability to generate almost real data. However, the original GAN is only suitable for processing continuous data such as images, while text is a typical kind of discrete data, so it cannot be applied directly to text generation. Many efforts have been made to adjust the internal calculation mechanism of GAN to deal with this problem, among which the introduction of RL greatly promotes the application of GAN in text generation. By combining the reward mechanism and the policy gradient technology of RL, GAN skillfully avoids the above problem that gradient cannot back propagation when facing discrete data and achieves promising results.

VAE and GAN are the two most powerful deep generative models which can generate data with complex distribution approximate to the real data distribution from random noise with simple distribution. The distinct difference between them is the distribution metric, that is, the loss function used to measure the quality of the generated data, is different. VAE utilizes an explicit measurement method that measures the KL divergence of training data and noise by assuming that training data is generated by another distribution. GAN, on the other hand, avoids the explicit measurement of distribution difference by making the neural network learning the measurement through adversarial training. In view of the two distribution measurement criteria are not perfect measurements, VAE and GAN have their own problems to be solved.

Faced with other problems of RNN, including the inability to effectively capture long-term dependencies, the vulnerability to the problem of gradient vanishing or exploding, and the lack of parallel computing capability, the Transformer model is proposed which adopts self-attention mechanism to replace the sequential structure in RNN. The self-attention mechanism can capture the context dependency among all words in a sequence to achieve more efficient sequence modeling without distance restrictions and obtain more semantically-rich text representations. Transformer has shown excellent performance in various NLP tasks since it was proposed and has great development potential.

In summary, these key techniques' advantages and disadvantages are compared in Table 1.

## 2.1 RNN

RNN is one of the most commonly used neural network models in text generation, whose natural sequence structure is suitable for the task of text sequence modeling. The recurrent structure in

Table 1. A summary of text generation techniques

Technique	Advantages	Disadvantages
RNN	Natural sequence structure is very suitable for the task of sequence modeling	Cannot effectively capture the long-distance dependence between sentences
GAN	Unsupervised learning; Generating clearer and more realistic samples than other generative models	Instable training process; Not suitable for processing discrete data, such as text
Reinforcement learning	Similar to human learning manners; Combining with GAN can subtly solve the existing problems in GAN and generate realistic text	Quite complicated training process
VAE	Leveraging the latent vectors to increase the diversity of the generated text	The latent variable ensures that the desired content is generated, regardless of its quality
Transformer	The attention mechanism can efficiently capture the long-term context information; Fast parallel computing speed	Large amount of calculation and slow training speed

RNN determines that it can process textual data sequentially, of which each hidden state takes the current input and the previous hidden state into consideration. Given the input text sequence  $X = (x_1, x_2, \dots, x_n)$ , the hidden state  $s_t$  of time step  $t$  is calculated as follows:

$$s_t = f(U \cdot x_t + W \cdot s_{t-1}) \quad (1)$$

After sequential processing, all semantic information of the given text are compressed into a fixed-length vector, the hidden state vector at the last time step, enabling the RNN model to have memory of previous content. Nevertheless, the problems of gradient vanishing or exploding still limit the application prospect of RNN. Variants of RNN model, such as long short-term memory (LSTM) and gated recurrent unit (GRU), combine the short-time and long-time memory through uniquely designed gating mechanisms, which makes them effectively solving these problems. There are three gate structures in LSTM that control the information in the cell state and selectively determine whether the information is retained or not. The forget gate determines what information in the cell state needs to be discarded according to the current input and the previous hidden state, which is calculated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

The input gate determines how much new information is added to the cell state, which is calculated as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

Then the cell state is updated based on the input gate and the forget gate as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (5)$$

Finally, the output gate determines the output of the LSTM unit at time step  $t$  according to the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (7)$$

Numerous researchers have shown that LSTM has the ability to generate natural and realistic texts in many generation tasks. Sutskever *et al.* [122] firstly propose the Sequence-to-Sequence (Seq2seq) learning model, which is a generalized framework for converting one sequence to another. In this framework, a LSTM as the encoder compresses sequences into vector representations. Then another LSTM as the decoder predicts output words one by one conditioned on the hidden state, and it takes the previous output as the input to predict the next output. Since this framework has no limitation of the length of input/output sequences, it has been widely used in text generation, including machine translation [22], text summarization [111], and dialogue system [129]. Consequently, the data-driven end-to-end training based on the Seq2seq model has become the mainstream method in text generation.

Actually, traditional Seq2seq models generally have two problems. The first is that all the inputs are transformed into a vector with fixed length, which limits the ability of latent vectors to represent input information, and the second is that assigning all the input words with the same weight cannot effectively capture the key information. To solve these problems, the Attention mechanism, an widely utilized mechanism in computer vision, is introduced into NLP. Through assigning different weights to different parts of the input sequence according to the current decoding state, the Attention mechanism can extract the key components from the input, which helps generation models make more accurate judgments while reducing the computation and storage consumption. The Attention mechanism is firstly applied to the Seq2seq model to fulfill machine translation tasks [5], and now has gradually become an important part of text generation models. For example, Xing *et al.* [141] introduce an attention-based multi-turn response generation model to capture the most relevant content in the conversation context. The Attention mechanism makes the multi-turn dialogue more coherent and consistent.

## 2.2 GAN and RL

From the perspective of neural network optimizing, there are still some defects in RNN-based generation models. First, most RNN-based text generation models are trained by maximizing the log-likelihood objective function, which may lead to the problem of exposure bias. Second, most loss functions are calculated at the level of words, while most evaluation metrics are based on the level of sentences, which may result in the inconsistency between the optimization direction of the model and the actual requirements. In this case, researchers introduce the GAN [47] into the study of text generation. GAN is composed of two parts: the generator and the discriminator. The generator produces false sample distributions similar to the real data, and the discriminator distinguishes generated samples and real samples as accurately as possible. For example, Zhang *et al.* [153] attempt to combine the LSTM and convolutional neural network (CNN) to generate realistic text using the idea of adversarial training.

However, the original GAN is only applicable to generate continuous data and has poor performance on processing discrete data because it is difficult for the gradient of the discriminator to correctly back-propagate through discrete variables. In order to solve this problem, Zhang *et al.* [153] utilize the smooth approximation algorithm to approximate the output of generator. Instead of utilizing the standard objective function of GAN, they match the feature distribution and make

the word predictions ‘soft’ in the embedding vector space to generate high-quality sentences. Researchers have also made some fine-tuning to GAN’s structure to generate discrete data, e.g., the Wasserstein GAN model [3].

Although the direct improvement of GAN has achieved some progress, it is still far from meeting practical requirements. Therefore, the idea of RL begins to be introduced to text generation. RL is usually a Markov decision process in which the action in each state will be rewarded (or reversely rewarded–punishment). For maximizing the expected rewards, the RL machine tries various possible actions in different states to evaluate the optimal policy according to the rewards provided by the environment. It can be seen that the reward mechanism in RL could help the GAN to deal with discrete data, which provides a new possibility for the application of GAN in text generation. For example, Yu *et al.* [148] propose the *SeqGAN* model to solve the problems of GAN in generating discrete text data. SeqGAN regards the text generation as a sequence decision procedure in RL, in which the generated sequence at each timestep represents the current state, the next word to be generated is regarded as the action to be taken and the returned reward is the discriminator’s score of the generated sequence. Through gradient policy algorithms, the SeqGAN model directly avoids the differentiability problem in the generator and obtain remarkable results in generating realistic natural language text.

### 2.3 VAE

Although the traditional Seq2seq model has made great progress in text generation, it tends to produce general and safe sentences with high probability, such as ‘I do not know’ and ‘I am sorry’. At the same time, the training of text generation system needs a large amount of high-quality labelled data, namely supervised training. However, in reality, most of the data is unlabelled and labelling a large amount of data is very time-consuming. The idea of unsupervised learning is introduced to solve this problem. VAE is a powerful unsupervised generative model, which contains an encoder that encodes input data into latent variables, and a decoder that decodes latent variables to reconstruct the original input data. Given the input  $x$ , the encoder will encode it into latent space  $p_{\theta}(z|x)$ , where  $\theta$  is the parameters of encoder. The decoder does the opposite which finds the probability distribution of the input based on the hidden variable  $p_{\phi}(x|z)$ , where  $\phi$  is the parameters of decoder. There is usually a latent hierarchical structure in natural language, and latent variables in VAE can capture and reflect such inherent language structure, so as to produce more natural and expressive natural language text.

The work of Bowman *et al.* [10] introduce a RNN-based VAE text generation model which assigns whole sentences with distributed latent vectors. By appending Gaussian prior distribution regularization on the hidden layers of the encoder, a sequence autoencoder model is constructed and the output sentence is generated word by word conditioned on the hidden vector to obtain consistency and diversity. Sequential data usually shows hierarchical structures and complicated dependencies between sub-sequences. For example, the sentence sequences and word sequences in a multi-round conversation have massive dependencies. Serban *et al.* [114] attach the latent variable to the hierarchical dialogue model to assign the generative model with multiple levels of variability for meaningful and diverse responses. Specifically They attach a high-dimensional latent variable to each sentence in the dialogue history, followed by generating responses conditioned on the latent variable.

### 2.4 Transformer

From the perspective of neural network training, RNN-based generation models also have some obvious shortcomings. Firstly, RNN processes the input sequence with strict linear order from forward to back, which leads to the problem of gradient vanishing or exploding due to the long

back propagation path. Secondly, RNN lacks efficient parallel computing capability due to its linear propagation structure where the calculation at the next time step relies on the outputs at the previous time steps. Therefore, RNN faces the issue of low calculation efficiency in large-scale application scenarios. To address this problem, Google proposes a new sequence modeling model, the Transformer model [126], which abandons the sequence structure in RNN and just contains Attention modules.

Specifically, the Transformer model is an encoder-decoder structure, only consisting of Attention modules and feed forward neural networks. The self-attention mechanism is the core of Transformer that captures the dependency among words in a sequence to obtain better semantic representations of each word. The self attention module is calculated as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (8)$$

The  $Q$ ,  $K$  and  $V$  are all vectors, each of which is obtained by multiplying the input vector by the weight matrix. The multi-head attention mechanism, composed of many self-attention modules to form an attention module, is proposed to further improve the ability of capturing context semantic information. After encoded by the self attention module, the output vector will be sent to the feed forward neural network, calculated as follows:

$$FFN(z) = max(0, ZW_1 + b_1)W_2 + b_2 \quad (9)$$

Besides the self attention module and the feed forward neural network module, there is also an encoder-decoder attention module in the decoder, which has the same mechanism as the traditional Seq2seq model. Due to the parallelization of the Attention module, Transformer has powerful parallel computing capacity and broad application prospect. Since Transformer was put forward, the various models based on it have achieved excellent performance in various NLP tasks, such as BERT[60] and GPT[106], making significant impact on the whole research area of NLP.

### 3 CONDITIONAL TEXT GENERATION

The development of deep neural networks brings unprecedented progress to text generation. However, there are still some problems with the existing text generation technology. For example, many studies train the text generation model only based on the content of input text, ignoring many other factors. However, a real person not only considers the context, but also adjusts the content according to their own conditions (such as mood and gender) and external factors (such as weather and environment) when speaking or writing. In this paper, we take conditional text generation (CTG) as the future research direction which is the key factor to improve the quality of generated text. Specifically, it includes context-based text generation, personalized text generation, topic-aware text generation, emotional text generation, knowledge-enhanced text generation, and visual text generation. In this section, we formalize the definition of CTG and introduce the wide application fields of it.

#### 3.1 The Concept Model

The CTG refers to taking certain external conditions into consideration to influence the generated results in the process of text generation. These conditions usually include *context*, *topic*, *emotion*, *external knowledge*, and so on. The general text generation methods only consider the text content factor, which makes the generated text less diverse and has a large gap with human expression. Consideration of external conditions in text generation makes it more anthropomorphic and brings better services to human beings in various fields.



We first give a formal definition of general text generation. Given the input text sequence  $X = (x_1, x_2, \dots, x_S)$ , the target of general text generation model is to generate the output text sequence  $Y = (y_1, y_2, \dots, y_T)$ , where  $S$  and  $T$  are the length of the input and output sequence respectively. The general text generation model can be defined as follows:

$$p(Y|X) = \prod_{t=1}^T p(y_t|X, y_{<t}) \quad (10)$$

1

At each decoding time step, the decoder will combine the input text and the output in the previous time steps to generate the current result. Finally, a text sequence with the highest probability is generated. For example, in the machine translation system,  $X$  might be a source Chinese sentence and  $Y$  might be an target English sentence, while in the dialogue system,  $X$  might be the query and  $Y$  might be the response, without considering other additional information in the generation process. On the basis of general text generation, the CTG models fuse additional condition to generate more anthropomorphic text. We define various kinds of CTG fields as follows:

*Definition 3.1. CONTEXT-BASED TEXT GENERATION: Integrating contextual information during text generation to enhance the understanding of environmental state to produce more coherent and informative text content.*

The contexts of natural language refer to the situations they are generated which are the key factor to ensure consistency and smoothness of the generated text. Given a set of contexts  $C = \{c_i\}_{i=1,2,\dots,K_c}$ , each context  $c_i$  may be a text sequence, a simple word, a sentiment score, and so on, and  $K_c$  is the number of context types. The context-based text generation model can be defined as follows:

$$p(Y|X, C) = \prod_{t=1}^T p(y_t|X, C, y_{<t}) \quad (11)$$

Take human conversation for example,  $C$  refers to the historical dialogue content in multi-rounds conversation. The daily dialogue process of human beings usually lasts for several rounds. The historical dialogue content in the multi-rounds dialogue is one kind of context information, and we will generate responses based on the historical dialogue to keep the conversation consistent. While in the advertisements writing scenario,  $C$  refers to various types of information about specific product, such as its brand, function, price, etc., and may also be information about user preferences. Only by integrating multi-kinds of contextual information can more appealing advertisements be generated.

*Definition 3.2. PERSONALIZED TEXT GENERATION: Assigning specific personalization characteristics to the text generation agents to produce personalized text contents which fit the given personalization characteristics.*

Personalization means that everyone has characteristics different from others, which will subtly influence how and what we express ourselves. Given a set of personalized characteristics  $S = \{s_i\}_{i=1,2,\dots,K_s}$ , each  $s_i$  represents age, gender, profession, or other characteristics, and  $K_s$  is the number of personalized characteristics types. The personalized text generation model can be defined as follows:

$$p(Y|X, S) = \prod_{t=1}^T p(y_t|X, S, y_{<t}) \quad (12)$$

Similarly, take dialogue as an example. People of different genders and ages have different views on the same thing, so the personalized characteristics  $S$  will have an impact on the dialogue content. When writing commodity description advertisements, it is necessary to combine the personalized features  $S$  of users, such as age, gender and shopping preference, so as to generate descriptions more in line with users' expectations. Therefore, in order to make the text generation agent more personified, it is necessary to assign specific personalized characteristics to generate text content conforming to the personalized information.

*Definition 3.3. TOPIC-AWARE TEXT GENERATION: Incorporating a specific topic in the process of text generation to make the whole text content suitable for the topic and ensure the coherence and rationality of the generated text.*

Natural language text has very strong internal relevance, especially in long text. A piece of text usually aligns around a specific topic, so considering topic information can generate more coherent and meaningful text. Given a set of topic words  $T = \{t_i\}_{i=1,2,\dots,K_t}$ , each  $t_i$  is a topic word and  $K_t$  is the number of topic words. The topic-aware text generation model can be defined as follows:

$$p(Y|X, T) = \prod_{t=1}^T p(y_t|X, T, y_{<t}) \quad (13)$$

When we write an article, we usually expand our thinking according to a specific topic  $T$ , such as maternal love, to ensure the logical consistency and consistency of the whole article. In the process of foreign language translation, it is necessary to combine the topic of the whole text content, to get a more fluent and consistent translation around the central topic. Combining topic information is a key factor to ensure logical coherence and compact semantics in text generation.

*Definition 3.4. EMOTIONAL TEXT GENERATION: Embodying the emotional expressions of the agents in the process of text generation, such as positive or negative, happy or sad, to adjust the content and expression style of the generated text.*

Emotion is a very important attribute of natural language, and people usually have certain emotions in daily communication or writing. Different emotions have important effects on what is being expressed. For example, when we are angry, we usually say something that is not rational or hurts others. Incorporating emotion into text generation can make the generated content more personified. In dialogue systems, it has a direct and quantifiable impact on product usability and user satisfaction when considering specific emotions. Given a specific emotion category  $E$ , which may be anger, sadness, joy, and so on, the emotional text generation model can be defined as follows:

$$p(Y|X, E) = \prod_{t=1}^T p(y_t|X, E, y_{<t}) \quad (14)$$

*Definition 3.5. KNOWLEDGE-ENHANCED TEXT GENERATION: Embracing external knowledge, such as search engine or knowledge base to provide factual basis and reference of the generated content in the text generation procedure.*

Human has a wealth of prior knowledge, and can flexibly combine our own knowledge in communication, translation or writing to express ourselves. Combining external knowledge in the text generation system can make the generated text more informative, more consistent with the logic of human expression, and reduce the possibility of common sense mistakes. Given a set of knowledge facts  $F = \{f_i\}_{i=1,2,\dots,K_f}$ , each  $f_i$  is a text sequence (also called free-text knowledge) or a

knowledge triple from knowledge graph (also called structured knowledge), and  $K_f$  is the number of knowledge facts. The knowledge-enhanced text generation model can be defined as follows:

$$p(Y|X, F) = \prod_{t=1}^T p(y_t|X, F, y_{<t}) \quad (15)$$

When we are asked a specific question, we will combine our knowledge to understand and reason the question, and find the corresponding answer to reply. For example, the question “What is the capital of China?” can be answered by combining geographical knowledge “Beijing is the capital of China”. The combination of knowledge is the key factor to ensure the real humanization of text generation system. Only through the interaction and expression of knowledge with the real world, can text generation agent be truly integrated into our daily life.

*Definition 3.6. VISUAL TEXT GENERATION: Integrating the semantic information in images into generated text, such as generating text descriptions according to image contents, or answering questions about given images.*

Data in our life is multi-modal, including not only text, but also images, sounds, and so on. We can automatically extract the information contained in the image and translate it into understandable natural language. Images can vividly depict external events and our psychological activities, so visual text generation has a rich application prospect. Given an image  $I$ , the visual text generation model can be defined as follows:

$$p(Y|X, I) = \prod_{t=1}^T p(y_t|X, I, y_{<t}) \quad (16)$$

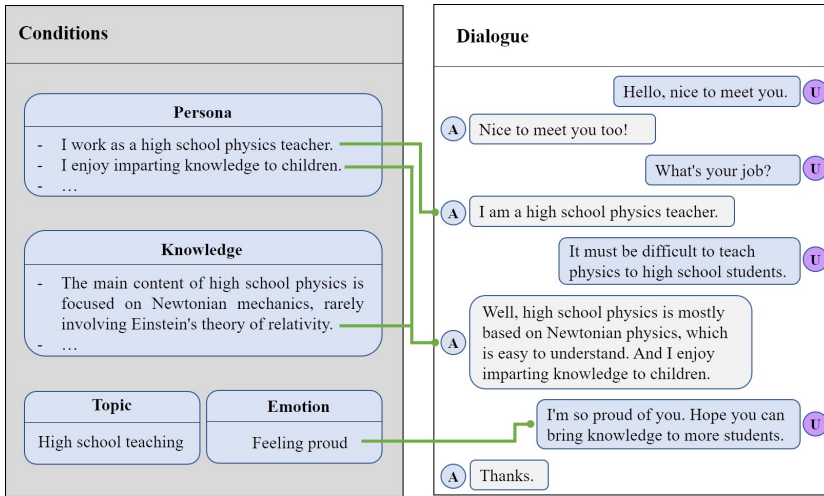


Fig. 1. The example of conditional dialogue system

Having formalized the various kinds of CTG, a real-life actual example, the dialogue system incorporating several conditions, is shown in figure 1. In this example, different conditions are considered to enhance the performance of the dialogue agent. The personalized characteristics  $S$  are two descriptive sentences about the work and preferences of the agent, and the knowledge facts  $F$  is a text paragraph about the content of physics in high school, while the context  $C$  is

the historical dialogue. The whole topic  $T$  of this conversation is “High school teaching”, and the emotion  $E$  inside the dialogue is “Feeling proud”. Based on these conditions, the dialogue agent can generate more relevant, personalized, substantial, and context-consistent responses.

We will discuss the technical details of implementing various kinds of CTG fields in the following sections. At present, most text generation models are based on the encoder-decoder structure, in which the encoder transforms input sequences into semantic vector representations, and the decoder generates outputs according to the input information, such as dialogue responses, and product reviews. Applying constraints of conditions in different parts of text generation models, including encoders, decoders, and their interaction modes, has been widely studied. In the encoding stage, external conditions can be encoded by various techniques, such as RNN and Transformer, and served as the input of the decoder together with the original input to control the generation process. The decoder can be modified by weighted decoding [46] or other technologies to control the decoding procedure to increase or decrease the probability of words with certain conditions. Meanwhile, the attention mechanism or RL can enhance the interaction mode of the encoder and decoder, to mine the implicit and deep semantic information in conditions. In short, only considering from multiple aspects including the encoder and decoder to integrate different conditions, can CTG systems produce content with higher quality and personification to provide us with more comfortably services.

### 3.2 Text Generation-based Human-Centric Services

Text generation technology has a wide range of application scenarios in daily life. It is an ongoing effort of the academic/industry researchers to use various text generation technologies to provide human-centric services, presented as follows.

**Goal-oriented dialog systems.** The dialogue systems are the most typical applications of text generation, which can be divided into goal-oriented and non-goal-oriented systems. Goal-oriented dialogue systems assist human to fulfil various tasks to reduce our operational burden, such as restaurant reservation, and travel time arrangement. In addition, goal-oriented dialog systems can also help companies accomplish specific businesses, such as customer transactions in a bank. According to Lauren Foye<sup>7</sup>, banks can automate up to 90% of their customer interaction using dialog systems by 2022. Apple Siri<sup>8</sup>, Microsoft Cortana<sup>9</sup>, Google Assistant<sup>10</sup>, and Amazon Alexa<sup>11</sup> are all typically frequently-used goal-oriented dialogue systems in our daily life. Siri is the first virtual assistant with a voice deployed in Apple devices to assist human to operate smartphones, which was born of this desire to make our interactions with computers more human-like, while Microsoft Cortana is the virtual assistant created by Microsoft for Windows 10, Windows Mobile and all of Microsoft’s integrated hardware. Google assistant is a smart personal assistant similar to Siri but can deploy on a wide range of devices, including android phones, android TV, wearable devices, etc. Take Siri as an example, we will introduce the detailed technical workflow of it.

On all IOS devices, we can say “Hey Siri” to invoke Siri hands-free. A very small speech recognizer runs all the time and listens for just those two words. When it detects “Hey Siri”, the rest of Siri parses the following speech as a command or query. The overall working flow chart of Siri is shown in the figure 2.

The input command of Siri passes through four stages for processing altogether. The first stage is speech recognition, where the “Hey Siri” detector uses a DNN to convert the acoustic pattern of

<sup>7</sup><https://www.juniperresearch.com/press/press-releases/chatbots-a-game-changer-for-banking-healthcare>

<sup>8</sup><https://www.apple.com/siri/>

<sup>9</sup><https://www.microsoft.com/en-us/cortana/>

<sup>10</sup><https://assistant.google.com/>

<sup>11</sup><https://developer.amazon.com/en-US/alexa/>

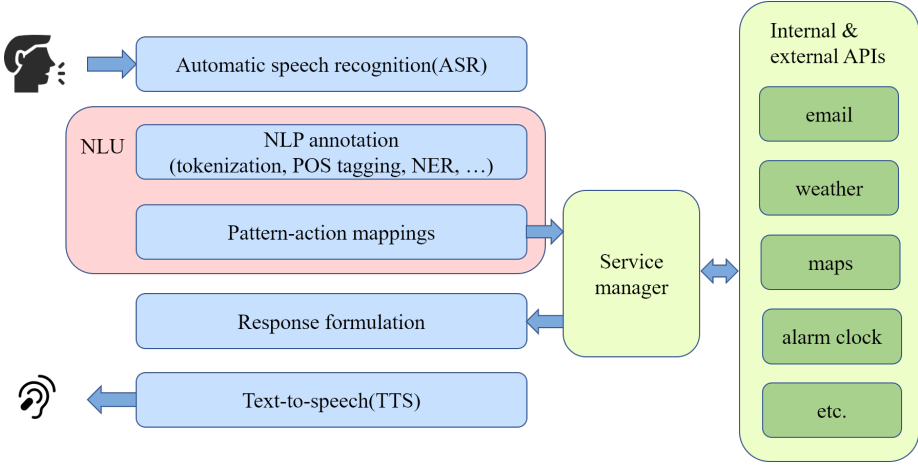


Fig. 2. The overall working flow chart of Siri

our voice at each instant into a probability distribution over speech sounds. It then uses a temporal integration process to compute a confidence score that the phrase we uttered is “Hey Siri”. If the score is high enough, Siri wakes up. After having collected and subsequently converted our command into a file, Siri sends it to Apple servers for processing. Once in the Apple servers, our spoken words undergo different flowchart branches to arrive at a possible solution. The third stage is to understand the meaning of the command using NLP technologies. Apple servers run NLP algorithms such as tokenization and named entity recognition, on input text to understand the intent of what the user is trying to say. For instance, the NLP engines can differentiate that when a user is saying “set an alarm for 7AM tomorrow”, the user is asking about setting an alarm and not about making a call. Finally, Siri communicates with other apps on the phone to provide the desired deliverable response to us in voice.

**Chatbots.** The non-goal-oriented dialogue systems, also known as chatbots, can communicate with humans normally in the open domain. Instead of completing specific tasks, chatbots engage in chatty conversations with humans and perform like a real person as much as possible. Chatbots provide us with a realistic and interactive dialogue experience and establish certain emotional connections with us. In recent years, with the emergence of a large amount of dialogue data and the breakthrough of machine learning applied to dialogue AI, intelligent dialogue systems have achieved gratifying results in the academia and industry. Microsoft XiaoIce<sup>12</sup> is one of the most popular social chatbots in the world that has made conversations with hundreds of millions of users and successfully built long-time emotional connections with them. Zhou *et al.* [165] describe the development of the Microsoft XiaoIce system to provide some guidance for chatbot researchers, which will be briefly summarized below.

XiaoIce is based on an empathic computing framework that enables chatbots to recognize human emotions and states, understand users’ intentions, and respond dynamically to users’ needs. Integration of IQ, EQ, and Personality is core to XiaoIce’s system design. IQ capacities include knowledge and memory modeling, image and natural language understanding, reasoning, generating, and predicting. These are the foundations for developing conversational skills that allow chatbots to meet the specific needs of users and help the user accomplish specific tasks. EQ refers to the need for chatbots to be emotionally intelligent enough to produce socially attractive

<sup>12</sup><http://www.msxiaobing.com/>

responses (e.g., having a sense of humor, comforting, etc.), and to be able to decide to drive a conversation to a new topic when it comes to a standstill, or to actively listen when the user engages in the conversation. Personality is defined as the characteristic set of behaviors, cognitions and emotional patterns that form an individual's distinctive character. Social chatbots need to hold a consistent personality and set the right expectations for the user during a conversation to gain users' long-term confidence and trust. The overall framework of XiaoIce is shown in the figure 3.

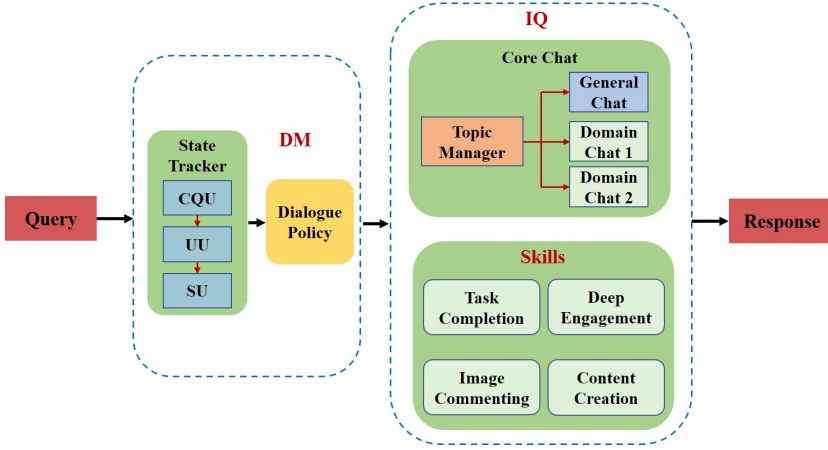


Fig. 3. The overall framework of XiaoIce

After receiving the information from users, the system will use Dialogue Management (*DM*) module to manage the whole dialogue process, in which the Global State Tracker is responsible for updating the system status. The Global State Tracker utilizes Contextual Query Understanding (*CQU*), User Understanding (*UU*), and System Understanding (*SU*) module respectively to integrate the dialogue context, users' characteristics, and system state for accurate system state capturing. Then the Dialogue Policy module decides the following dialogue strategy according to the updated dialogue status, that is, whether the query is to be answered by the Core Chat or a certain skill.

The Core Chat is designed for open domain conversation, which is divided into General Chat and Domain Chat. General Chat mainly answers general questions, while Domain Chat focuses on answering professional questions in the particular domain. The main realization technique of chitchat is the combination of retrieval model and sorting model, in which the retrieval model generates candidate response sets, and the sorting model sorts the candidate responses to output the response with the highest score. In addition to the Core Chat, XiaoIce also has a number of constantly updated skills, including Image Commenting aiming to generate comments based on user input images, Content Creation accomplishing creative work such as poetry creation and story generation.

**Question Answering.** In addition to the dialogue systems, the Question Answering (*QA*) system, providing corresponding answers to users' different questions, is another typical application of text generation. The *QA* system needs to find relevant content through search engines or knowledge bases to organize the corresponding answers which may relate to commonsense facts or details about specific events. Dehghani *et al.* [31] propose a *QA* model, called *TraCRNet*, to achieve the goal of open-domain query answering. The *TraCRNet* model reasons to correctly answer the question by utilizing information of multiple documents extracted from a search engine which includes not

only the high-ranked web pages but also the low-ranked web pages that are not directly related to the question.

**Machine translation.** With the development of economic globalization, the world has become a small village where people from all over the world can communicate with each other via the Internet. As a result, translation becomes an essential requirement for better communication with each other. Machine translation systems also provide researchers with the opportunity to exchange ideas with researchers around the world, enabling scientific researches to develop more rapidly and vigorously. Google translate<sup>13</sup> is a free translation service offered by Google. It provides instant translation between 80 languages and supports the translation of words, sentences, and web pages between any two languages. According to statistics, Google translate translates over 100 billion words every day, which is one of the most popular translation software in the world. The Google Translation team puts forward the Google Neural Machine Translation system (GNMT) [139] for the first time in 2016, which consists of a deep LSTM network with 8 encoder and 8 decoder layers using residual connections as well as attention mechanism. The model architecture of GNMT is shown in the figure 4.

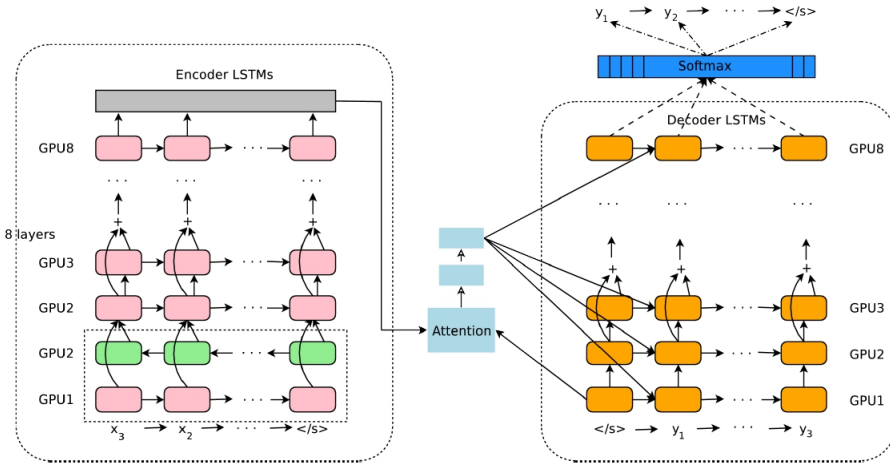


Fig. 4. The model architecture of GNMT

The GNMT is a typical sequence-to-sequence learning framework with the attention mechanism, composed of an encoder, a decoder, and an attention network. The encoder encodes the input sentence into vector representations, and the decoder generates one word at a time according to the current states. The attention mechanism allows the decoder to focus on the important information in the source sentence to build the connection of the encoder and decoder. With a large amount of paired translation data for training, GNMT can achieve excellent translation performance. In the meanwhile, to solve problems in neural network model training, Google proposes a lot of training tricks to improve the performance of the model. The low-precision arithmetic is employed in the inference computations to accelerate the final translation speed, the length-normalization procedure and coverage penalty are employed in the beam search technique to encourage the output sentence to cover all the words in the source sentence. The words in the input and output are divided into a limited set of common sub-word units (“wordpieces”) to balance the flexibility of

<sup>13</sup><https://translate.google.com/>

“character”-delimited models and the efficiency of “word”-delimited models for improving handling of rare words.

After the GNMT system, Google proposes the Transformer model for the machine translation task, which completely abandons the common network structure in RNN, and only adopts the attention mechanism to carry out sequence modeling. After that, Transformer has become a standard structure for many NLP tasks, giving a big boost to the development of the NLP field.

**Product review and advertisement generation.** Due to the large number of product reviews in the shopping websites, writing reviews for products may puzzle customers and waste their time. Fortunately, based on the information of a given product and the rating of the review, reviews can be automatically generated by the review generation technology, providing references for other customers. Ni *et al.* [100] build an assistant system for helping users to write reviews. This model expands the contents of input phrases and conforms to users’ personalized aspect preferences to generate diverse and smooth product reviews. At the same time, writing specific advertisements for vast products is also a time-consuming task, particularly when we want to generate personalized ads for each consumer. The personalized advertisement generation technology can automatically generate well-suited product description according to different selling points and user preferences/traits. It not only provides great convenience for consumers, but also for sellers. Chen *et al.* [18] propose a personalized product description generation model by leveraging neural networks combined with the knowledge base.

**Text summarization.** In recent years, the volume of text data from various sources has exploded. Text summarization systems help people quickly understand the main content of information and improve the efficiency of information acquisition. According to Radeff *et al.* [105], a summary is defined as “a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that”. The purpose of a text summarization system is to produce a concise and fluent summary of the source articles while retaining key information content and overall meaning. Nallapati *et al.* [99] firstly introduce the attentional encoder-decoder architecture into text summarization system, and achieve the state-of-the-art performance, which provides the direction for the follow-up research works.

**Data storytelling.** There is a lot of structured data in real life. Computers are good at analyzing structured data, while humans are more inclined to read complete stories rather than a jumble of data. Therefore, how to make use of structured data to create more personified stories is a key research direction of text generation. Data storytelling is the key technology to achieve this goal, whose typical applications include news report generation, company report generation, sports event report generation, and so on. News report generation systems can produce complete, meaningful, and reasonable news according to the time, place, cause, process, and other factors of the specific news event, while sports report generation systems can generate detailed game reports according to the situation of sports events. Narrative Science<sup>14</sup> is a text report generation company whose purpose is to give life to data. It is dedicated to the research of automatic text generation technology, which can automatically generate a high-quality text after a planning process based on the key information in the data and its expression in the machine.

**Image captioning and visual QA.** Text information is just one way for human to obtain information, while we are more likely exposed to image information in real life. Image captioning technology can automatically generate corresponding text description according to the content of images, so as to facilitate readers to better understand image contents. Feng *et al.* [39] train

<sup>14</sup><https://narrativescience.com/>



an unsupervised image captioning model to generate image captions without the paired image-sentence datasets. Visual QA is another interactive point between text generation and image understanding. By understanding image content and corresponding questions at the same time, visual QA technology can generate answers of questions related to the images. Li *et al.* [70] convert visual QA question into a machine reading comprehension problem combined with the large-scale external knowledge base to realize the knowledge-based visual QA.

With the rapid development of NLP technology, it is ubiquitous to find the usage of text generation technology in our daily life. Nevertheless, text generation has been far from mature. For example, it is still easy to find out whether we talk to a real person or a chatbot, which means that an obvious gap still exists between robots and human beings. CTG is the key factor to solve this problem, which aims to generate the high quality and anthropathic textual content. In this paper, we will study different types of conditional text generation fields.

## 4 MAJOR RESEARCH AREAS

After introducing the definition and application scenarios of CTG, in this section, we make a detailed investigation of different CTG fields, including context-based text generation, personalized text generation, topic-aware text generation, emotional text generation, knowledge-enhanced text generation and visual text generation. Furthermore, summarize the multi-conditional text generation and pre-trained language model-based CTG works.

### 4.1 Context-based Text Generation

In many applications of text generation, the context information is the key factor to realize the coherence and smoothness of the generated text. The context means the situations in which natural languages are generated. In dialogue systems, the context usually refers to the dialogue history that has taken place in multi-rounds dialogues. The ability to consider previous utterances is the core to build active and engaging dialogue systems [16]. Meanwhile, in review generation systems, the context refers to the time, emotions, sentiments and other factors. The context information provides clues to the generation of natural language [123]. Therefore, in order to generate high quality text, it is necessary to consider the context information in CTG. We give a brief summary of context-based text generation methods in Table 2.

**Context embedding.** The most simple and effective way to combine context information is to embed it directly to get the vector representations as a part of the decoder input, so that context information can be considered in the generated text. Among the numerous tasks of context-based text generation, the utilization of context in dialogue systems attracts great attention of researchers. For instance, Sordani *et al.* [121] embed all words and phrases in the dialogue history into vector representations. Dialogue history information is encoded into vectors, which are decoded by another RNN to produce context-aware responses. In addition to the dialogue system, other applications also need to add different types of context information. Tang *et al.* [123] define the *context* as the information or situations that may influence the output content. The encoder encodes the contexts information (*e.g.*, a sentiment score, a product id or a user id) into continuous semantic representations and concatenates them as the input of the decoder. During decoding, the context information are attended through a gating mechanism to generate context-sensitive product reviews. Similarly, Clark *et al.* [24] propose a text generation model in stories, which treats entity representations extracted from dialogue history as context. By encoding the historical conversations together with the entity representations as context, the model can better determine which entities or words to be mentioned next.

When considering the context information in machine translation systems, multiple sentences can be treated as a whole, and relevant information between sentences can be captured, which

Table 2. A summary of context-based text generation methods

Work	Method	Description
Sordoni <i>et al.</i> [121]	Context embedding	Embedding all the words and phrases in the dialogue history into continuous representations as additional inputs of the decoding stage
Voita <i>et al.</i> [131]	Context embedding	Encoding the source and the context sentence separately and learning the context-aware representation of the source sentence through the attention mechanism
Serban <i>et al.</i> [113]	Hierarchical context embedding	Embedding the word sequences in each context sentences at the low-level and embedding the sentence sequences in the historical dialogues at the top-level to efficiently capture the context information
Xing <i>et al.</i> [141]	Hierarchical context embedding	Leveraging the attention mechanism to extend the HRED model
Jaech <i>et al.</i> [56]	Context Adaption	Using the context information (dialogue history) to transform the weights of recurrent units in RNN to effectively capture high-dimensional context

not only prevents errors in the case of ambiguity, but also improves the consistency of translation. Voita *et al.* [131] propose a context-aware machine translation model that can control and analyze the flow of information from context to the translation model. The source sentence needed to be translated and the context sentence are encoded separately to get the context-aware representation of the source sentence through the attention mechanism. It is identified that pronoun is the key information captured by the model. In the field of machine translation, Kang *et al.* [58] proposed to select contextual sentences dynamically for each source sentence to be translated. The Context Scorer module is used to score each context sentence based on the currently translated source sentence and incorporate important context sentences into the translation module.

**Hierarchical context embedding** Instead of embedding context information directly, hierarchical context embedding method divides the embedding process into two steps to capture information in context more effectively. Concretely, the first step is to embed the word-level information, and the second step is to embed the sentence-level information. Serban *et al.* [113] use Hierarchical Recurrent Encoder-Decoder (HRED) model to hierarchically encode the dialogue history and guide the generation of replies. In particular, the word sequences in each context sentence are encoded at the low-level, while the sentence sequences in the historical dialogues are encoded at the top-level. Xing *et al.* [141] leverage the attention mechanism to extend the HRED model. By incorporating attention mechanism at the words and sentences level respectively, the model captures the most important parts in the context. Zhang *et al.* [150] observe that we can have more smooth conversations without much context information in the multi-user dialogue, and produce a tree-based hierarchical multi-user dialogue model, which builds a tree structure consisting of many branches for multi-user conversations to select exact context sequences. Besides, Tian *et al.* [124] conduct a comprehensive survey on existing context-aware conversational models and find that compared

with the non-hierarchical model, the hierarchical model is more capable for capturing context information.

**Context adaption.** The context adaption method changes the model itself rather than regarding context as an extra input of CTG model. For example, Jaech *et al.* [56] utilize the context information to transform weights of the recurrent layer in RNN. In particular, they utilize a low-rank decomposition algorithm to control the degree of parameter sharing in context, which performs well on high-dimensional and sparse context.

**VAE-based methods.** How to ensure the coherence of context in the generation of long-form text (e.g., a single or multiple paragraphs) is a challenging problem. Both high-level abstract features (e.g., topics, sentiments, etc.) and low-level fine-grained features (e.g., specific word choices) should be considered to generate globally-coherent long text sequences. Traditional RNN models tend to generate repetitive and inconsistent long-form text due to the poor feature extraction ability, while VAE models based on deep latent variables can capture these high-level features to generate coherent and high-quality long text. For example, Shen *et al.* [117] propose a VAE-based multi-level network structure for CTG, which contains a multi-level decoder to capture coherent long-term structure inherent in long-form text by generating high-level intermediate representations of input sentences. Meanwhile, multiple stochastic layers are used between VAE's encoder and decoder to generate more semantically-rich latent variables for producing more coherent and less repetitive long text. Shao *et al.* [116] firstly design a sequence of groups, and subsequently generates each sentence conditioned on the planning result and previously generated context sentences. A hierarchical latent structure containing global planning and local sequence latent variables is used to improve the diversity of the generated text.

## 4.2 Personalized Text Generation

Various human characteristics significantly impact interpersonal communication and writing styles. In other words, personalization plays a key role in enhancing the quality of CTG model. In dialogue systems, personalization is vital for creating truly smart dialogue agents which can be seamlessly incorporated into the lives of human beings. In product review generation systems, personalization ensures the generated product review depends not only on the attributes of the product, but also on the preferences of specific users, endowing the authenticity for the generated reviews. Many efforts for personalized text generation are conducted, as summarized in Table 3, and we will discuss them in details below.

**Personalized feature embedding.** The simplest method to achieve personalized text generation is to embed the personalized characteristics of different users. Li *et al.* [71] present a speaker model which encodes user profile into vectors so as to capture personalized features and guide the response generation during the decode stage. Instead of encoding personalized features into vector representations directly, Herzig *et al.* [52] use an additional neural network to capture the high-level personalized information based on the personality traits. The additional layer implicitly influences the decoding hidden state to ensure that the personalized features are integrated into the generated text. Li *et al.* [72] propose the User-aware Sequence Network (USN), to generate a summary for a user's review according to his preference on different aspects or writing style. The user-aware encoder selects the user-concerned information in a review, and the user-aware decoder combines user characteristic and user-specific language habits into word generation. Zheng *et al.* [159] propose a trait fusion module to capture the persona information of each speaker. Each persona trait is encoded into vector representation and all traits are merged to get the integrated persona vectors. The persona aware attention mechanism controls the attention weights of the context vector and the persona-aware bias estimates the word generation distribution. Luo *et al.* [86] build a personalized goal-oriented dialog system for the restaurant reservation task. The profile

Table 3. A summary of personalized text generation methods

Work	Method	Description
Li <i>et al.</i> [71]	RNN + Speaker model	The speaker model encodes each individual speaker into a vector to capture characteristics; Generating personal responses matching a specific user
Luan <i>et al.</i> [84]	Autoencoder + Multi-task learning	Training a response generation model on a small personalized dialogue data, and then training an autoencoder model with non-conversational data; Sharing parameters of the two models to obtain the personalized dialogue model
Yang <i>et al.</i> [145]	Transfer learning + Pretrain and fine-tuned	Respectively using massive generic dialogue data and a small-scale personalized dialogue data to pre-trained and fine-tune the dialogue model to generate personalized responses
Yang <i>et al.</i> [144]	Reinforcement Learning + Persona embedding	Embedding user-specific information into vector representation; RL mechanism optimizes three rewards – topic coherent, informative and grammatical, to generate more personalized responses

model encodes user profiles into vector representations and stores conversation history from similar users. The preference model captures user preferences over knowledge base entities and combines with the profile model to enhance the performance in terms of task completion and user satisfaction. To improve the personality consistence of the generated dialogue responses, Song *et al.* [118] propose the generate-delete-rewrite mechanism to delete inconsistent words from a generated response prototype and further rewrite it to a personality-consistent one.

**Multi-task and transfer learning.** The personalized text datasets are so scarce that the above models are difficult to perform very well. Some researchers attempt to enhance the performance of personalized text generation by transfer learning and multi-task learning models. For instance, the work of Luan *et al.* [84] trains a dialogue model to predict responses given previous contexts and an autoencoder model with large volumes of non-conversational personal data to model the role-specific characteristics of different users. Through the multi-task learning mechanism which shares the decoder parameters of the two models, these models can capture speaker roles, expressive styles and domain expertise characteristic of the targeted user and generate personalized responses without heavy recourse to each speaker's conversational data. Yang *et al.* [145] propose a domain adaptation-based personalized dialogue model. They respectively use massive generic dialogue data and a small-scale personalized dialogue data to pre-trained and fine-tune the dialogue model, and apply the policy gradient algorithm to improve the personalized and informative features of generated responses. Similarly, Zhang *et al.* [152] put forward the Learning to Start (*LTS*) model to optimize the quality of responses, which divides the training process into initialization (modeling the responding style of human) and adaptation (generating personalized responses) for generating relevant and diverse responses.

**GAN and RL models.** Our writing style can be perceived by his specific word usage manners, which means different language habits can reflect our personalized characteristics. Yuan *et al.* [149] propose a personalized sentence generation model based on GAN. In the training procedure, the frequently used words are incorporated as the input sources. Then the sentence structure is constrained to generate sentences similar to the original sentences of the same author. RL can control the quality of generated content through different policies or rewards, so researchers consider incorporating it to implement personalized text generation. Yang *et al.* [144] present the attention-based hierarchical encoder-decoder architecture via RL to realize personalized dialogue generation, which defines three types of reward mechanisms, including *topic coherence*, *mutual information*, and *language model* to force the text generation model to generate topic-relevance and coherent dialogue responses.

**VAE-based methods.** The above embedding based personalized text generation methods learn user information from training data and cannot discover the common properties among users. User-level features can also be depicted through latent variables, so VAE-related models are introduced into personalized text generation. Wasserstein Auto-Encoders (WAE) is a typical VAE variation, which conducts adversarial training on latent variables, instead of assuming that latent variables subject to a simple Gaussian distribution to fit the real data distribution and improve the generation ability of VAE. Chan *et al.* [14] embed and mix the user-level and sentence-level information into multimodal latent distributions. The mixed distribution is then regarded as the prior distribution of WAE, and extended to the Gaussian Mixture Distributions to guide the decoder to generate personalized responses for different users. To generate product tips with personalized features of users, Li *et al.* [77] present the Persona-Aware Tips Generation model (PATG), which employs adversarial variational auto-encoders to model the persona information of different users. The persona information is distilled from all historical tips and reviews of a target user and expressed by the latent variables in VAE. An external memory-based Pointer Network is also deployed to conduct the memory reading to retrieve more accurate persona information.

### 4.3 Topic-aware Text Generation

Topic information is indispensable in our daily communication, reading or writing. We usually have a conversation around a specific topic and usually identify the topic of an article before we read or write it. In this subsection, we give a review of topic-aware text generation studies, as presented in Table 4.

**Topic extracting and embedding.** It is a common idea to extract topic information from existing text and embed it into vector representations to guide text generation. Xing *et al.* [140] propose a topic-aware Seq2seq (TA-Seq2Seq) model to generate informative and interesting responses for chatbots. TA-Seq2Seq incorporates topic information of the dialogue history extracted by the pre-trained LDA model with the input sentence, and utilizes the joint attention mechanism to guide the generation process. Choudhary *et al.* [23] observe that topic information can be divided into multiple domains (e.g., games, sports or movies) to provide the fine-grained guidance for the generator. They adopt domain classifiers to capture domain information from the dialogue history for generating domain-relevant responses. Feng *et al.* [38] develop a multi-topic-aware LSTM (MTA-LSTM) model to generate a paragraph-level text under target multiple topic words. In the MTA-LSTM model, each topic will be assigned with different weights to maintain a multi-topic coverage vector, which is updated in the decoding process in order. Then the vector will guide the generator to generate the topic-aware text with an attention module. A long article usually spans many topics, while a simple text summary usually cannot cover all topics. To generate text summaries of specific topics of interest to users, Krishna *et al.* [67] propose to generate multi summarizations for a given article according to different topics. With an article and a topic of

Table 4. A summary of topic-aware text generation methods

Work	Method	Description
Xing <i>et al.</i> [140]	RNN + LDA + Topic embedding	Utilizing LDA to get topic information, and embedding the topic words into the vector; Generating more informative, and topic relevant responses
Dziri <i>et al.</i> [35]	HRED + LDA + Topic embedding	Combining topic and context information to produce not only contextual but also topic-aware responses
Wang <i>et al.</i> [133]	CNN + LDA + RL	Using LDA to get topic information, CNN to capture the dialogue information, and RL to optimize the model with specific evaluation metric; Generating coherent, diverse, and informative text summaries
Feng <i>et al.</i> [38]	RNN + Topic embedding	Assigning each topic with different weight to maintain a multi-topic coverage vector and updating them in the decoding process in order

interest as input, the proposed pointer-generator network will pay higher attention to the relevant parts of the topic in the input article to generate topic-tuned summarizations. Dziri *et al.* [35] introduce a Topical Hierarchical Recurrent Encoder Decoder (*THRED*) model to generate contextual and topic-aware responses. *THRED* hierarchically encode the dialogue history in the word and sentence level respectively and capture topic information from dialogue context using a pre-trained LDA model. In the decoder, the generation probability is biased towards generating topic words by adding an extra probability to the original generation probability.

**CNN-based methods.** The above studies mostly employ the RNN model in the task of topic-aware text generation. In addition to RNN, several other models are also applied to this task, which also achieve remarkable results. For instance, Wang *et al.* [133] propose a topic-aware convolutional Seq2seq (*ConvS2S*) model, which leverages the joint attention and biased probability generation mechanism for incorporating topic information. Word and topic embeddings of the source sequence are encoded by the associated convolutional blocks. Then the joint attention mechanism attends to words and topics according to the decoder states, and the biased probability generation is performed to generate coherent, diverse, and informative summaries,

**VAE-based methods.** The latent variables in VAE can capture features implicitly in natural language, which is useful for providing high-level guidance to text generation. If latent variables correspond to a specific topic, the probability distribution of generated words will be narrowed down, thus improving the rationality of generated content. Consequently, the VAE is widely used in topic-aware text generation. For instance, Wang *et al.* [135] propose a topic-guided variational autoencoder (*TGVAE*) method to generate natural language text under the guidance of the designated topic. Specifically, *TGVAE* generates a Gaussian mixture model (*GMM*) for latent variables as the prior distribution which is parametrized by a neural topic module responsible for capturing long-range semantic information in the whole document. Each mixture component corresponds to a specific topic, which guides to generate semantically-meaningful sentences under the given topic. Gao *et al.* [42] propose a neural variational language model to study the topic-level Gaussian distributions in latent space. They utilize CNN to get the vector representations of input sentences

Table 5. A summary of emotional text generation methods

Work	Method	Description
Zhou <i>et al.</i> [162]	GRU + Emotional embedding	Emotion category embedding captures emotional information and the internal emotion memory balances the grammaticality and the expression degree of emotions
Fu <i>et al.</i> [40]	GRU + Multi-task learning	Multi-decoder Seq2seq module generates outputs with different styles and style embedding module augments the encoded representations
Kong <i>et al.</i> [66]	Conditional GAN (CGAN) + Sentiment control	The generator generates sentimental responses based on a sentiment label and the discriminator distinguishes the generated replies and real replies
Li <i>et al.</i> [74]	RL + Emotional editor	The emotional editor selects the template sentence based on the topic and emotion, and RL forces the model to enhance the coherence and emotion expression of generated responses

and predict the Gaussian distribution of topics using the full connection layer. By sampling the topic distribution, the proposed model can generate diversified sentences conditioned on given topics. In previous works of text generation based on VAE, the distribution of latent variables is usually assumed as Gaussian distribution, which makes it difficult to distinguish which part of latent variables controls the structure and which part controls the semantics of natural language. In order to solve this problem, Li *et al.* [75] develop the *TATGM* model, which adopts a sequential VAE to learn the structural features of text and a topic model to extract the semantic features of text to generate different expressions of the same structure in different topics. *TATGM*'s topic model generates text based on the Gaussian distribution of latent variables, which ensures the capture of textual semantic information. At the same time, the encoder acts as a discriminator to force the decoder to generate the text with similar semantics.

#### 4.4 Emotional Text Generation

Natural language is full of emotions, and emotional words are more likely to stimulate the interest of readers. Additionally, people adjust their speaking style and content according to their own and other people's emotional changes in daily communication. Due to the necessity of integrating emotional information, researchers pay attention to incorporating emotional information into the generated text in order to provide users with better experience, as summarized in Table 5.

**Emotion extraction and embedding.** One common way to generate emotional text is to extract emotional information from input text and embed it into vector representations as the input of the decoder. Asghar *et al.* [4] propose an LSTM-based emotional dialogue generation model with three designed mechanisms to incorporate affective/emotional aspects into generated responses. The affective word embeddings introduce an external cognitively engineered affective dictionary to augment traditional word embeddings with affective vectors. The affective loss functions minimize the Euclidean distance between the affective embeddings of inputs and responses, and maximize the

affective content of responses to explicitly train an affect-aware model. The affectively diverse beam search injects affective dissimilarity across the beam groups based on affective word embeddings to promote the generation of emotionally rich responses. Zhou *et al.* [162] produce the Emotional Chatting Machine (*ECM*) to generate grammatically correct, context-relevant and emotionally consistent responses. The *ECM* leverages emotion category embedding for capturing high-level abstraction of emotion expressions, an internal emotion state for balancing grammaticality and emotion dynamically, and an external emotion memory to help generate more explicit and unambiguous emotional expressions. Majumder *et al.* [88] consider that emotional responses often mimic the emotion (positive or negative) of the user. The emotion stochastic sampling and emotion mimicry mechanism are proposed to encode context and emotions to generate appropriate and empathetic responses.

**Emotion transferring.** In addition to embed emotional information directly, the transfer from one emotion to another is a promising way to generate emotional text. Fu *et al.* [40] achieve the goal of transferring the emotion of reviews from positive/negative to negative/positive through multi-task learning and adversarial training. They leverage a style embedding module to augment the language style representations and a multi-decoder Seq2seq model to respectively generate text with different styles. Luo *et al.* [85] propose the *Seq2SentiSeq* model, which adopts the Gaussian Kernel layer to incorporate the numeric sentiment intensity value into the decoder, so as to finely control the sentiment intensity of generated text. At the same time, the cycle reinforcement learning algorithm controls the process of model training which balances both sentiment transformation and content preservation through the elaborately designed rewards to tackle the problem of lacking parallel data.

**GAN and RL models.** Several works confirm that GAN and RL have significant effects on emotional text generation. For example, Wang *et al.* [132] propose the SentiGAN with multiple generators and one multi-class discriminator to enhance the sentiment accuracy and quality of generated texts. In the SentiGAN, multiple generators are trained simultaneously to generate texts with different sentiment labels, such as positive or negative, and the multi-class discriminator makes each generator focus on generating its own examples of a specific sentiment label accurately. Kong *et al.* [66] introduce a conditional GAN (*CGAN*)-based sentiment-controlled dialogue generation model. The generator of *CGAN* produces sentimental responses under the given dialogue history and sentiment label, while the discriminator identifies the quality of generated responses through checking whether the items (*e.g.*, dialogue history, sentiment label, and dialogue response) belongs to the real data distribution. Li *et al.* [74] propose the emotional editor module to select the template sentences according to emotion and topic information in the dialogue history, and introduce RL to promote the quality of generated responses from three points: emotion, topic and coherence.

**VAE-based methods.** Utilizing latent variables of VAE to control the sentiment of generated text has also been explored by researchers. For example, Hu *et al.* [55] combine VAE and holistic attribute discriminators for effective imposition of semantic structures. They allocate one dimension of the latent representation to encode “positive” and “negative” semantics, to capture a salient attribute independent with other features. The global discriminators facilitate effective imposition of latent code semantics to guide the discrete text generator learning. Chen *et al.* [17] endow the poetry generator with the ability to express the specific sentiment (*e.g.*, negative and positive), to improve the semantics and diversity of generated poems. Since sentiments are often strongly coupled with semantics in poetry, the authors make latent variables conditioned on both sentiment and text content to capture generalized sentiment-related semantics. Besides, a temporal sequence module captures sentiment transition patterns among different lines of the poetry to generate diverse poems under the control of semantic-level and line-level sentiments.



Table 6. A summary of knowledge-enhanced text generation methods

Work	Method	Description
Ghazvininejad <i>et al.</i> [45]	Keyword matching + Facts embedding	The Facts Encoder module leverages an external memory for embedding the conversation-related facts to generate content-rich responses
Dinan <i>et al.</i> [32]	Transformer + Memory Network	Memory Network retrieves knowledge about the dialogue from the memory and Transformer encodes and decodes the text representations to generate responses; Conducting knowledgeable discussions on open-domain topics
Zhou <i>et al.</i> [163]	Knowledge graph attention	Graph attention mechanisms integrate commonsense information from the knowledge based on the dialogue history; Generating more appropriate and informative responses
Mazumder <i>et al.</i> [93]	Lifelong learning + Open-world knowledge base completion	Obtaining new knowledge by asking users when facing unknown concepts and then inferencing to grow knowledge over time

#### 4.5 Knowledge-enhanced Text Generation

Nowadays, most text generation systems take advantage of deep neural network models to generate fluent, semantic and consistent text. However, there is still a big gap between such machine-generated text and human expression, that is human will combine their knowledge in speaking or writing, while most text generation systems fail to achieve this. By combining sufficient knowledge, such as commonsense knowledge and information about specific objects/events, CTG systems can generate more logical, credible, and informative text. The external knowledge includes structured knowledge graph (KG), which is composed of knowledge triples with the form of  $\langle head, relation, tile \rangle$ , and unstructured knowledge base (KB), which is composed of natural language text about specific concepts. There are many ways to combine external knowledge in CTG systems, as summarized in Table 6, and we will introduce them in details below.

**Unstructured KB enhanced models.** Unstructured KBs contain abundant knowledge with textual form. How to extract the knowledge related to input and combine it into the generation stage are main directions of current research. The common way to extract knowledge from unstructured KB is keyword matching with each word in the input as keywords. For instance, Ghazvininejad *et al.* [45] extract relevant knowledge facts from knowledge base using keyword matching and encode them into vectors to provide factual evidence for the dialogue response generation. In order to extract relevant knowledge more effectively, Ren *et al.* [110] propose to use the global perspective for selecting appropriate knowledge. A topic transition vector is obtained from the context to express global information and then used to guide the local knowledge extraction process for generating informative and fluent text. Zhao *et al.* [157] separate parameters relying on knowledge-grounded dialogues in the whole model to get better results with insufficient knowledge-grounded dialogue data. In the decoder stage, the *Language Model* generates common words, the *Context Processor* module generates context words, and the *Knowledge Processor* module generates words from knowledge document by a hierarchical attention mechanism. The decoding manager fuses the

generation probability of three modules, and dynamically switches the decoding mode according to decoding states, to generate context-relevant, informative and reasonable responses.

After extracting the relevant textual knowledge, the next most important thing is to understand its semantics and integrate it into the text generation process. Young *et al.* [147] transform the extracted knowledge triples into a sequence tokens and encode them into vector representations using LSTM. The context vectors and knowledge vectors are concatenated to calculate match scores with different responses to select the most appropriate response. Wang *et al.* [136] build a technical-oriented dialogue system to communicate with people about Ubuntu-relevant questions. The knowledge text descriptions are embed into vectors by word embedding average or BERT model which are concatenated with traditional word embeddings to enhance the understanding of the technical term in dialogue history. A knowledge reader attentively read knowledge embeddings and retrieve the semantic information at each decoding stage to generate informative responses. In addition to RNN, Dinan *et al.* [32] combine *Transformer* and *Memory Network* to build an open-domain knowledge-based dialogue system. The Memory Network retrieves related knowledge from the Internet according to the input as the knowledge memory. Each sentence in the memory is independently encoded with a Transformer encoder, and the same Transformer is used to encode the dialogue context. The standard dot-product attention between the memory candidates and the dialogue context is performed to select knowledge sentences to be used which are served as the input of the decoder to generate knowledgeable responses.

Due to the powerful performance of Transformer, more and more researchers begin to use it for knowledge understanding. Zhao *et al.* [156] introduce the hierarchical interaction between the context and external document knowledge to capture the most important parts in the document and context using the multi-head attention module in Transformer for selecting the most appropriate response. Li *et al.* [78] generate vector representations of external knowledge using the multi-head attention mechanism and then incorporates them to encode knowledge utterances span in the multi-turn dialogue. The decoder firstly generates contextual coherence responses attending on the context information and then refines them by attending on the knowledge vectors to generate more informative response. In knowledge-enhanced dialogue systems, there can be one-to-many relations between the dialogue context and the knowledge, which makes the knowledge selection is diverse and difficult. Kim *et al.* [62] propose to keep track of the prior and posterior distribution over knowledge using laten variables to improve the knowledge selection accuracy. Through sequentially modeling the history of knowledge selection in previous turns, the scope of probable knowledge candidates at current turn is reduced. The posterior distribution over knowledge leverages the response information to select knowledge more accurate.

**Structured KG enhanced models.** Knowledge graph is a kind of structured knowledge base, which describes physical entities and their connections accurately. As its name suggests, knowledge graph is also helpful to build knowledge-enhanced text generation systems. Wang *et al.* [134] propose an entity linking module to decide the optimal entity in the input question for selecting knowledge triples from external KG, which will be encoded as common words by the LSTM. The similarity scores of the question and relation candidates are calculated to select the most appropriate triples for question answering. The TranE algorithm [9] is proposed to transform structured triples into low dimensional vector representations and is widely used in knowledge-enhanced text generation systems. Moussallem *et al.* [98] link knowledge facts based on the translated document, encode them into vectors by TransE, and concatenate them with the internal vectors of NMT embeddings as the decoder input to enhance the quality of generated translations. Gune *et al.* [49] extract entities from external KG and adopt TransE to get vectors of them. The knowledge vectors are then fed into the separate multi-head attention channel to generate coherent text summaries. The attention mechanism is also applied in learning knowledge form knowledge graphs, which is

called graph attention. Guan *et al.* [48] present an incremental encoding schema to mine hidden information in the story context and graph, and a contextual attention mechanism to encode knowledge graph into vectors. The multi-source attention mechanism is used to comprehensively understand the content of stories to generate reasonable and consistent story endings. Zhou *et al.* [163] produce a knowledge-based dialogue model (CCM) that leverages two graph attention mechanisms to promote dialogue understanding and knowledgeable responses generating. The *static graph attention* module encodes the graphs relevant to the dialogue history and concatenate graph vectors with input vectors to enhance the semantic information of the input. The *dynamic graph attention* module attentively reads all the knowledge graphs and all triples in each graph based on decoder states to adaptively choose a generic word or an entity from the retrieved graphs for word generation.

To generate text with more entities, Moon *et al.* [97] propose the *DialKG Walker* model that learns the symbolic transitions of dialog contexts as structured traversals over KG, and the *graph decoder* that attends on viable KG paths to predict the most relevant entities in the KG, by associating these entities with the dialogue context and entities mentioned in the previous turn. Koncel *et al.* [65] study the problem of generating paragraphs with multiple sentences given only a short title. The *Graph Transformer* model computes the hidden representations of each node in a graph by attending over its neighbors following a self-attention strategy to leverage the relational structure of knowledge graph. The decoder attends on encodings of the knowledge graph and document title using the decoder hidden state to generate informative texts. Chen *et al.* [19] propose a data-to-text generation model, which extracts entities appear in the data field and links them to Wikidata as external knowledge to form the temporary memory. The dual attention mechanism is applied to generate words conditioned on both input table information and background knowledge fact information. To consider the dialogue context in the knowledge retrieve process, Wu *et al.* [138] design a Felicitous Fact mechanism to help the model focus on the knowledge facts that are highly relevant to the context.

GCN [64] is an extension of CNN in the graph domain, which can effectively learn the structural information of nodes and edges in the knowledge graph. De *et al.* [30] regard the question answering problem as the graph inference problem. Nodes in this graph correspond to named entities in a document whereas edges encode relations between them (e.g., cross and within-document coreference links or simply co-occurrence in a document). The entity graph relates mentions to entities within and across documents, the document encoder obtains representations of mentions in context, and the relational GCN propagates information through the entity graph to generate correct answers.

**Continuous learning models.** Although existing studies introduce some real-world knowledge to CTG systems, the knowledge is usually fixed and cannot be expanded or updated. Continuous learning in the interactive surroundings is an important capability of human beings. We keep on learning and updating our knowledge according to our experiences in the daily life, which should be considered as an important factor when building humanoid text generation systems. Mazumder *et al.* [93] build a knowledge learning model, namely lifelong interactive learning and inference (*LiLi*), enabling chatbots to interactively and continuously learn new knowledge when communicating with users. By mimicking humans to acquire knowledge, *Lili* enquires ask users for related items when facing unknown concepts and then infers to expand knowledge over time.

#### 4.6 Visual Text Generation

Since people usually gather information from images, visual text generation is also an important research direction in text generation. Two of the most important applications are image captioning and visual QA. A summary of visual text generation methods is given in Table 7.

Table 7. A summary of visual text generation methods

Work	Method	Description
Vinyals <i>et al.</i> [130]	RNN + CNN	Encoder CNN captures information in images, and decoder RNN generates neural language descriptions based on their features
Malinowski <i>et al.</i> [89]	LSTM + CNN	CNN and LSTM respectively encodes the image and the question into vectors to capture the semantic information, and then another LSTM generates corresponding answers
Dai <i>et al.</i> [26]	Conditional GAN (CGAN)	CNN captures information in an image, and LSTM generates the relevant descriptions; the discriminator evaluates the quality of generated descriptions
Das <i>et al.</i> [28]	Memory Network	Embedding the image, historical dialogue and given question respectively to consider the image and dialogue context information in conversation

**Combining CNN and RNN.** In order to capture the information in images and generate natural language text, combining CNN and RNN is the main solution for visual text generation. Vinyals *et al.* [130] achieve the goal of automatically viewing an image and generating the reasonable description utilizing the encoder-decoder structure. The encoder CNN captures information in the image, and the decoder RNN generates the text description. Due to the heavy loss of image information causing by the high-dimension structure of CNN, Xu *et al.* [142] propose an image caption model utilizing attention mechanism to extract the most important information in images to generate more accurate and detailed image description. Malinowski *et al.* [89] combine CNN with LSTM to answer questions about the given image. They utilize the CNN to capture the related information in the image about questions, and an LSTM to generate answers based on latent representations of the image and question. Zhu *et al.* [166] build a semantic relationship between text descriptions and regions in the image by object-level grounding to generate answers of questions correspond with specific image regions.

**Memory Network-based models.** Instead of simple single-round visual QA, Das *et al.* [28] implement a visual dialogue system to communicate with users in multiple rounds about a given image. They put forward the task of *Visual Dialogue* and publish a large-scale Visual Dialogue dataset called VisDial<sup>15</sup>. Three novel encoders are designed for the visual dialogue task, in which the *Late Fusion* module encodes the image, historical dialogue and the given question respectively, the *Hierarchical Recurrent Encoder* module encodes the dialogue history in the sentence level and the *Memory Network* module stores the former QA pair as the “fact” to offer factual basis for the latter responses generation.

**GAN-based models.** Different from the above works, Dai *et al.* [26] leverage the conditional GAN (CGAN) model to generate high-quality image descriptions in three aspects, including naturalness, semantic relevance, and diversity. They jointly learn a generator to produce descriptions conditioned

<sup>15</sup><https://visualdialog.org/>

on images and an evaluator to assess how well a description fits the visual content with the criteria of natural and semantically relevant.

**VAE-based methods.** In image captioning, it is important to ensure the lexical and syntactic diversity of generated captions. Chen *et al.* [15] propose the Variational Multi-modal Inferring tree (*VarMI-tree*) to model the lexical and syntactic diversities by inferring their latent variables in an approximate posterior inference guided by a visual semantic prior. Conditioned on the visual features and the latent variables, diverse captions of given images are generated. Previous works usually generate latent variables for entire input sentences, ignoring information about the substructures in the sentences. Aneja *et al.* [1] develop the *SeqCVAE* model, which learns a latent space for every word to capture the ‘intention’ about how to complete the sentence. The data-dependent transition model captures the ‘intention’, a representation of the remaining part of the sentence by encoding them with a backward LSTM, to generate more diverse captions.

#### 4.7 Multi-conditional text generation

We have summarized existing works of CTG under a single condition, but in practice, these conditions often act simultaneously on the text generation system to produce more reasonable and anthropomorphic content. For example, in daily conversations, generation models usually consider the context information, personality characteristics of the interlocutor, and abundant external knowledge to generate reasonable responses. Therefore, considering the hybrid of different conditions is essential for improving the quality of CTG systems.

Emotion is an inherent attribute of natural language. Explicit emotion modeling and combining other conditions can improve the naturalness and humanness of the generated content. In order to realize emotional text generation, it is necessary to first detect the emotions contained in the textual content. Based on the observation that people usually express emotions rely on conversation contexts and external knowledge, Zhong *et al.* [160] propose to interpret the contextual utterances and leverage the external commonsense knowledge to enhance the emotion detection performance. The hierarchical self-attention and cross-attention modules are used to abstract contextual information and the context-aware affective graph attention is used to leverage knowledge to facilitate the understanding of context and emotion detection in conversations. Kao *et al.* [59] develop the chatbot that will change emotions according to the conversation context with users. Through the sentiment recognition model, the dialogue agent can provide the robot’s emotions as feedback while talking with a user. Peng *et al.* [103] believe that topics, like emotions, are important factors in dialogue systems, which can ensure the semantic coherence of the whole conversation. They develop the Twitter Latent Dirichlet Allocation (*LDA*) model to detect the topic words of the input sentences as the prior knowledge, and the dynamic emotional attention mechanism to obtain the content and affective information related to the input texts and additional topics. In order to generate recommended reason text of specific items for users in recommender systems, Bai *et al.* [7] fuse aspect sentiment and external knowledge for recommended reasons generation. The fine-tuned BERT model is applied to get the aspect and aspect sentiment polarity from the reviews. The aspect fusion module fuses aspects and the item title, and the knowledge fusion module fuses relevant knowledge by the bi-directional self-attention module to generate personalized and content-rich recommended reasons. Zhong *et al.* suggest that persona plays an important role in empathetic conversations, and first present a novel large-scale multi-domain dataset for persona-based empathetic conversations [161]. Based on this dataset, they propose an efficient BERT-based response selection model, CoBERT, using multi-hop co-attention to learn higher-level interactive matching.

External knowledge can provide guidance for any CTG system to improve the system’s understanding of conditions and generate more informative texts. For instance, Chen *et al.* [18] introduce

a Knowledge Based Personalized (*KOBE*) product description generation model which fuses product aspects, user categories, and knowledge base to generate informative and personalized product descriptions. The self-attention modules in Transformer are used to encode the product attributes, the relevant knowledge, and the specific user categories into semantic vector representations, and perform deep semantic interaction to capture semantic features for the decoder. Yang *et al.* [143] fuse external knowledge into topic-to-essay generation systems to provide background information for essay generation. The memory-augmented neural model selects knowledge concepts and then stores them into a memory matrix. The decoder then attends the memory to guide the text generation and updates it according to the decoder states to generate topic-consistent and informative essays.

The generation of natural language text is usually influenced by many factors, so the combination of multiple conditions is a promising research trend of CTG systems. By considering the appropriate context, combining accurate knowledge, expressing specific emotions, and conforming to unique personalized characteristics, text generation systems can generate more anthropomorphic texts.

#### 4.8 Pre-trained language model-based CTG

The idea of pre-training has been widely explored in NLP. By pre-training the model on large-scale text corpus to initialize most of the network parameters which learn universal knowledge about syntactic and semantic information of neural language text, and fine-tuning the model using a small amount of specific downstream task data, excellent performance can be achieved. The pre-trained language models are first introduced into NLP for word embedding [96]. Using a large amount of data to train the LSTM language model in an unsupervised way, the contextual word vector of each word can be obtained to demonstrate strong results across discriminative natural language understanding (*NLU*) tasks [94] [104].

Recent pre-trained language models based on large Transformer architectures prove the ability of both big models and big data to improve language representation and generation performance. Transformer is gradually replacing the mainstream position of LSTM in NLP. Among numerous works, the BERT model [60] and the GPT model [106] receive the most attention. BERT learns bidirectional representations of massive textual data by conditioning on both the forward and backward sequential contexts. Just adding a specific output layer rather than adjusting the model's structure, the pre-trained BERT model can be fine-tuned to achieve the state-of-the-art performance in many NLU tasks, such as text classification. Subsequently, a lot of work is done to optimize the pre-training process of BERT to further improve the ability of language representation, among which the most typical work includes XLnet [146], RoBERTa [81], and ELECTRA [25].

In terms of unconditional natural language generation, the GPT models are regarded as starting points for pre-trained natural language generation due to its form of standard language model for text generation tasks. GPT adopts the typical pre-training and fine-tuning training framework, with Transformer decoder as the feature extractor. In the pre-training stage, the training task is the unidirectional language model to encode language knowledge into the decoder, while in the fine-tuning stage, parameters of the pre-training model are fine-tuned according to specific tasks. Considering that supervised fine-tuning of the pre-training model with data of specific domains will reduce the generalization ability of the model, GPT-2 [107] and GPT-3 [11] remove the supervised fine-tuning stage of GPT and directly use massive training samples for zero-shot training. For all NLP tasks, GPT-3 with 175 billion parameters shows impressive performance without any gradient updating or fine-tuning.

Designing a pre-trained model for natural language generation tasks which often adopt Seq2seq frameworks with attention mechanism, is highly potential and critical. In addition to the GPT series that only contain Transformer decoders, researchers also explore the Seq2seq pre-training

for unconditional text generation to jointly train encoders and decoders for better generation performance, including MASS [120], UniLM [34], BART [68], T5 [108] and so on. For example, MASS combines the pre-training of encoder and decoder to reconstruct a sentence fragment, which masks a piece of tokens of input sentences randomly in the encoder and predicts the masked tokens in the decoder. The joint training process improves the ability of feature extraction and language modeling, which can achieve promising generative performance with zero-shot or few-shot fine-tuning on task-specific data. More and more researches show that pre-trained language models are of great value in text generation tasks.

After exploring the great potential of pre-trained language models for text generation, CTG powered by pre-trained language models has become a promising research direction. Based on the powerful generation ability of pre-trained language models, text generation models can be better at controlling the expression of conditions and generating more personified text under specific conditions. The most straightforward way to incorporate pre-trained language models to CTG systems is to modify the model architecture for extra conditional inputs or condition-specific fine-tuning. Mao *et al.* [90] perform intermediate fine-tuning on the story data to adapt the pre-trained GPT-2 model to the domain of stories, and then fine-tune on the target story generation dataset with a multi-task learning objective to generate grammatical and stylistic consistency stories. An auxiliary training signal is used to provide common sense grounding for generated stories, which constrains the model to rank sensible text with lower perplexity. Keskar *et al.* [61] propose to add a mention, namely control codes specific to each type of text (e.g. “books” for novel-type texts), to the input text and include one at the beginning of each text during the pre-training phase. By this means, the Conditional Transformer Language (CTRL) model learns the relationship between the control codes and the text that follows to determine the generated text under the desired condition controlled by the specific code. Considering that large pre-trained transformer models are sensitive to large parameter changes during fine-tuning, Ziegler *et al.* [167] choose to adapt the pre-trained language model to arbitrary conditional inputs. The pseudo self-attention module learns a task-specific encoder which injects learned encoder conditioning directly into the pre-trained self-attention of the model. Because of the arbitrary length of the input sentence, these additional conditional inputs can be injected into the pre-trained model without changing the model architecture to affect the generated text. Chen *et al.* [21] leverage the idea of model distilling for better text generation performance. The Conditional Masked Language Modeling (C-MLM) task is proposed to enable pre-trained BERT with additional conditional input by randomly masking tokens only in the target sequence. In the knowledge distillation stage, the generated sequences of word logits by the *teacher* BERT model contains information from both backward and forward contexts, providing sequence-level global guidance. This probability distribution is soft targets for the *student* text generation model to mimic, which contains more useful and fine-grained conditional information. To leverage the redundant external knowledge under capacity constraint, Zhao *et al.* [158] propose a pre-trained language model-based response generation model with a knowledge selection module, which formalize knowledge selection as a sequence prediction process. The knowledge selection and response generation module are jointly optimized with unlabeled dialogues to endow a generative model with both rich knowledge and good generalization ability.

Above CTG works based on pre-training language models need to adjust the model structure or fine-tune the model with the data under specific conditions, entailing the significant cost of retraining. The Plug and Play Language Model (PPLM) [29] allows anyone to flexibly plug in one or more simple attribute models representing the desired control objective into a large, unconditional LM. Instead of training a large language model from scratch, PPLM trains smaller attribute models to influence the generated results of the existing ones, such as GPT-2. Attribute models are responsible for estimating the probability that a text sequence  $x$  with a specific attribute  $\alpha$  (e.g. Positive or

Negative). By maximizing the probability of the generated sequence  $x$  with the desired attribute  $\alpha$ , the generated sentence has the pre-defined conditions. Considering that PPLM still requires large amounts of labeled texts to effectively balance generation fluency and proper conditioning, Carbone *et al.* [12] leverage topic models to enhance PPLM with an unlabeled collection of documents. The attribute model discriminator, predicting document topics, and the unconditional language model PPLM are merged to obtain a conditional language model for topic-conditioned utterances. To equip the dialogue model with multiple skills (*e.g.*, emphatic response, weather information, etc.) without retraining the whole dialogue model, Madotto *et al.* [87] propose the Adapter-Bot, which triggers different skills via different Adapters trained independently. The backbone of the Adapter-Bot is a pre-trained conversational model such as DialoGPT [154], providing the ability of response generation. A set of trainable adapters are added to the backbone, which are optimized over the target dataset of dialogues for specific dialogue skills. Using the trained dialogue manager to select the right dialogue skill under the dialogue story, Adapter-Bot shows high-level control over the chatbot.

To generate text under more precise conditions in the word-level and phrase-level rather than just high-level conditions such as topic and sentiment, Chan *et al.* [13] propose the Content-Conditioner (CoCon) model to control the generated text under the guide of target text content at a fine-grained level. The encoder and decoder of the model are pre-trained by GPT-2, and the CoCon block incorporates the target content into the encoded text representation before passing the content-conditioned representations into the decoder for generation. By self-supervised training without labeling data, CoCon can produce high-quality text with content. Zhang *et al.* [155] explore the problem of generating text from a given set of lexical constraints. Given lexical constraints, the proposed *POINTER* model generates high-level words (*e.g.*, verbs and adjectives) as the keywords constrains, then inserts other words of finer granularity around the keywords iteratively until the whole sentence is generated. The training objective of *POINTER* is to generate a complete text sequence with a set of keywords as constraints, which is similar to the masked language modeling (MLM) objective in BERT, so pre-trained BERT is used to initialize the model training to boost the generation performance.

The latent variables in the VAE model can capture higher-level sentence representations, such as topics, semantics, and patterns, to facilitate CTG. In order to combine VAE's powerful ability in the pre-training language model, Li *et al.* [69] propose the OPTIMUS model, the first large-scale pre-trained deep latent variable models. OPTIMUS includes two stages: pre-training and fine-tuning. In the pre-training stage, the sentence-level (variational) auto-encoder objectives are trained on large text corpus to construct a universal latent space for reorganizing sentences. In the fine-tuning stage, by representing labeled specific tasks' data in the pre-trained latent space, OPTIMUS fine-tunes the latent space by updating all/part of the parameters to adapt to the downstream tasks. Specially, for the stylized response generation task, through embedding the history, response, and style-reference into the joint latent space to fine-tune OPTIMUS, the generated responses are closer to the desired text style.

With sufficient pre-training on huge text corpora, large pre-trained language models have shown impressive language understanding and language generation ability. Researches of CTG leveraging pre-trained language models have also attracted more and more attention, as summarized above. How to integrate conditional information into pre-training language models more effectively and reduce the cost of retraining is the important future research direction.

#### 4.9 Decoding Strategies for CTG

We have made a detailed investigation of different CTG fields above, such as context-based text generation and topic-aware text generation. Meanwhile, the hybrid of different conditions and



pre-trained language models applied in CTG systems are also been discussed. In summary, previous works focus on modifying the encoder, or the interaction mode between encoders and decoders to fuse conditional information into the text generation process. In the decoding stage, different decoding strategies can have a huge impact on the quality of the generated text. Researchers have proposed several decoding strategies to improve the quality of generated contents, *e.g.*, reducing repetitive words or phrases. Through straightforwardly restricting the probability distribution of different words during the decoding stage under specific conditions, high-quality content with specific conditions can be generated by CTG models. In this section, we will introduce several universal decoding strategies, such as Beam Search and Top-k Sampling, and then summarize some improved decoding strategies proposed for CTG, including Weighted Decoding and Unlikelihood Training.

**Beam Search.** Due to the large number of words in the vocabulary, the number of possible sequences in generation is enormous, so researchers propose some heuristics to reduce the searching space thus making the generation practical. Beam Search is a most commonly used decoding strategy in text generation models recently. It approximately maximizes the likelihood of the whole generated sequence given a hyper-parameter  $\beta$ , known as beam size. In the first decoding time step,  $\beta$  words with the highest conditional probability are selected as the first words of the candidate output sequence. For each subsequent time step, additional words will be attached to  $\beta$  output sequences of the last time step, in which the  $\beta$  sequences with the highest conditional probability will be selected. The optimal generated sequences are determined from the final  $\beta$  candidates with the highest conditional probability. Beam Search greatly reduces the time and space requirements for searching in the text generation process by limiting the beam size. However, since only  $\beta$  sequences are selected at each time step, the final generated content may not be optimal.

To overcome shortcomings of Beam Search, many improved methods have been proposed. For example, Vijayakumar *et al.* [128] propose the Group diverse Beam Search to increase the variety of generated texts, which divides the beam into groups and utilizes a group dissimilarity penalty to reduce the similarity between different search groups. Similarly, Li *et al.* [73] propose the Sibling diverse Beam Search that contains a penalty proportional to the rank of a candidate token to encourage preserving hypotheses from diverse sources within the beam.

**Top-k and Top-p sampling.** The goal of sampling-based decoding strategies is to reduce repetitions and increase the diversity of the generated content, by utilizing stochastic decisions in the generation process. The Top-k sampling [37] samples from the next-token distribution after having filtered this distribution to keep only the top  $k$  tokens, while the Top-p sampling [53], also known as nucleus sampling, samples from the top tokens with a cumulative probability just above a threshold  $p$ .

The above decoding strategies are aiming at making the distribution of generated words similar to the distributions of words in human-generated texts at a higher level. In the meanwhile, many studies focus on decoding strategies serving for specific conditions, to make CTG systems more effectively fuse the conditional information, which will be discussed below.

**Weighted Decoding.** Weighted Decoding [46] is a typical decoding strategy that increases or decreases the probability of words under certain conditions. Ghazvininejad *et al.* [46] propose an interactive poetry generation system which enables users to edit and polish generated poems by adjusting from different aspects (*e.g.*, sentiment, alliteration, etc.). In the  $t^{th}$  decoding step, a partial hypothesis  $y_{<t} = y_1, \dots, y_{t-1}$  is expanded by calculating the generation score for next word  $w$ :

$$score(w, y_{<t}; x) = score(y_{<t}; x) + \log P_{RNN}(w|y_{<t}, x) + \sum_i w_i * f_i(w; y_{<t}; x) \quad (17)$$

$\log P_{RNN}(w|y_{<t})$  represents the generative log-probability of the word  $w$  and  $score(y_{<t}; x)$  is the accumulated generation score of all generated words. The  $f_i(w; y_{<t}; x)$  refers to decoding features with corresponding weights  $w_i$ . A decoding feature assigns a real value to the generation probability of word  $w$ , to control the weights of generated words under different conditions. Weighted Decoding considers 8 types of features to control the generated content, including topical words, emotions, and so on. Instead of manually designing calculation formulas of decoding features as additional items of the decoding objective function, Holtzman *et al.* [54] train a number of discriminative models to construct a more powerful generator under different aspects of conditions, each of which encodes an aspect of high-quality generation to enhance the generation performance of the original RNN generator. The decoding objective function is formalized as follows:

$$f_{\lambda}(x, y) = \log(P_{lm}(y|x)) + \sum_k \lambda_k s_k(x, y) \quad (18)$$

This objective is composed of the traditional RNN language model probability  $\log(P_{lm}(y|x))$  and additional scores  $s_k(x, y)$  calculated by designed discriminators with learned mixture coefficients  $\lambda_k$ . Four discriminators are proposed to discriminate between good and bad generations (e.g., Repetition Model for avoiding word repetitions, Relevance Model for guaranteeing contextual relevance of the generated content, etc.) and are interpolated in the objective function as log probabilities. The weight coefficient of each discriminator is optimized to minimize the difference between the scores assigned to the gold continuation and the continuation predicted by the current model. Similarly, Baheti *et al.* [6] incorporate topic and semantic similarity constraints into the decoding objective of dialogue systems to encourage the generation of more topic-relevant and content words in responses. In order to match the distribution over topics in generated responses and input, the *HMM-LDA* model is applied to estimate topic distributions over words in responses and inputs so as to compute their topical similarity. The semantic similarity constraints are designed to encourage generated responses to be semantically similar to inputs.

Weighted Decoding is a useful technique in conditional text generation, which can force the expected conditional features to appear in the generated text by assigning them a high generation probability. However, when the weight of the specific feature is too large, Weighted Decoding risks going off-distribution, thus generating unanticipated words [112].

**Unlikelihood Training.** The standard approach of training neural text generation models is to maximize log-likelihood and approximately decoding the most likely sequence, which is known to have fatal defects. The likelihood training will force the model to generate common words with high frequency, making the generated text dull, and to repeat themselves at word and sentence level. In order to solve these problems, Welleck *et al.* [137] optimize the *unlikelihood training* objective for training text generation models more efficiently. The key idea of unlikelihood training is to decrease the generation probability of certain tokens, making incorrect repeating tokens less likely and frequent tokens less likely, thus forcing repetitions to have low probability and improving the quality of generated text.

Considering that existing dialogue systems tend to generate frequent words and repetition words, Li *et al.* [76] first incorporate unlikelihood training into dialogue systems to regularize generated outputs to match human-written distributions. Three different biases needed to be mitigated are considered, including repetition and copying, vocabulary usage, and contradictions. The first two biases are mainly responsible for reducing the occurrence of repeated words and high-frequency words, and the bias of contradictions is designed to assign low probability to inconsistent and contradictory utterances. Through dividing training samples into positive and negative coherent behavior, the likelihood training is performed on coherent data, and the unlikelihood objective

is applied to the incoherent data to reduce the probability of generating the context incoherent response.

Through unlikelihood training, text generation models can force unlikely generations to be assigned lower probabilities, such as repetitive and dull text generation, to improve the overall quality of generated sentences. In summary, unlikelihood training has great potential for researches of CTG systems. By controlling the probability of specific conditional features expressed in the generated content, the efficient fusion of conditional information, and higher-quality conditional text generation can be achieved.

#### 4.10 Conditional datasets

The training of CTG models needs the support of a large number of conditional text data, such as emotional text data or personalized text data, but this kind of data is relatively scarce. In order to protect researchers from data scarcity, many high-quality conditional text datasets have been released. We give a brief summary of conditional text datasets in Table 8.

**Context-based datasets.** Cornell Movie Dialogs<sup>16</sup> [27] is a large-scale multi-turn dialogue dataset providing contextual information during the conversation, which contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts. Ubuntu Dialogue Corpus<sup>17</sup> [82] is another large multi-turn dialogue dataset containing almost one million conversations extracted from the Ubuntu chat logs that is a great help for training context-sensitive technical dialogue systems.

**Personalized datasets.** Zhang *et al.* [151] present a high-quality personalized dialogue dataset named PERSONA CHAT<sup>18</sup>. In each dialogue, two parts of the conversation are given a group of profile information, and the whole dialogue process is conducted around these personalized characteristics. Humeau *et al.* [92] build an authoritative profile-based dialogue dataset using conversations collecting from REDDIT<sup>19</sup>. The personalized characteristics are extracted from users' social posts, providing a new opportunity of personalized text generation for later researchers. Chen *et al.* [18] provide a personalized and knowledge-based product description dataset named TaoDescribe, collecting from Taobao<sup>20</sup>, a large Chinese shopping website. There are more than two million pairs basic information and descriptions of products, in which each pair is labeled with knowledge and user category attributes.

**Emotional datasets.** Rashkin *et al.* [109] publish an emotional dialogue dataset called *Empathetic Dialogues*<sup>21</sup>, including an extensive set of emotions and every speaker in it feels with a given emotion during conversations. Chen *et al.* [20] publish another high-quality emotional dialogue dataset collecting from telescripts and dialogues in Facebook, named *EmotionLines*<sup>22</sup>. All utterances in it are labeled with specific emotion according to textual contents to guide the emotional dialogue response generation.

**Knowledge-based datasets.** CMU DoG<sup>23</sup> [164] is a document grounded conversation dataset where each conversation is followed by specified documents about popular movies extracted from Wikipedia articles. Wizard of Wikipedia<sup>24</sup> [32] is another open-domain dataset whose conversations are directly grounded with knowledge retrieved from Wikipedia. Zhou *et al.* [163] present a

<sup>16</sup>[http://www.cs.cornell.edu/~cristian/Cornell\\_Movie-Dialogs\\_Corpus.html](http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html)

<sup>17</sup><https://github.com/rkadlec/ubuntu-ranking-dataset-creator>

<sup>18</sup><https://github.com/facebookresearch/ParlAI/tree/master/projects/personachat>

<sup>19</sup><https://www.reddit.com/r/datasets/comments/3bxlg7/>

<sup>20</sup><https://www.taobao.com/>

<sup>21</sup><https://github.com/facebookresearch/EmpatheticDialogues>

<sup>22</sup><http://doraemon.iis.sinica.edu.tw/emotionlines/index.html>

<sup>23</sup><https://github.com/festvox/datasets-CMUDoG>

<sup>24</sup>[https://parl.ai/projects/wizard\\_of\\_wikipedia/](https://parl.ai/projects/wizard_of_wikipedia/)

Table 8. A review of conditional text datasets

Dataset	Type	Description
Cornell Movie Dialogs [27]	Context-based dataset	A multi-turn dialogue dataset extracted from raw movie scripts
Ubuntu Dialogue Corpus [82]	Context-based dataset	A multi-turn dialogue dataset extracted from the Ubuntu chat logs
PERSONA CHAT [151]	Personalized dataset	A personalized dialogue dataset where two parts of every conversation are given a group of profile information
Humeau <i>et al.</i> [92]	Personalized dataset	A profile-based dialogue dataset; Extracting personalized characteristics from users' posts in REDDIT
TaoDescribe [18]	Personalized dataset	A personalized product description dataset including product basic information, in which each pair is labeled with knowledge and user category attributes
Empathetic Dialogues [109]	Emotional dataset	An emotional dialogue dataset where each conversation is under a given emotion
EmotionLines [20]	Emotional dataset	An emotional dialogue dataset in which all utterances are labeled with emotions
CMU DoG [164]	Knowledge-based dataset	A document grounded conversation dataset where each conversation are about contents of a specified document
Wizard of Wikipedia [32]	Knowledge-based dataset	A knowledge-grounded dataset with conversations directly grounded with knowledge retrieved from Wikipedia
Zhou <i>et al.</i> [163]	Knowledge-based dataset	A commonsense conversation dataset containing one-turn post-response pairs with corresponding commonsense knowledge graphs
VisDial [28]	Visual dataset	A Visual Dialogue dataset where all queries and answers are based on the given image

commonsense conversation dataset<sup>25</sup> containing one-turn post-response pairs with the corresponding commonsense knowledge graphs. Each pair in it is associated with some knowledge graphs retrieved from ConceptNet<sup>26</sup>, a typical structured knowledge graph.

**Visual datasets.** VisDial<sup>27</sup> [28] is a large-scale visual dialogue dataset where all queries and answers are based on the given image. Researchers can utilize it to research the visual text generation.

#### 4.11 Evaluation Methods Towards CTG

We have given a detailed investigation of different CTG fields and summarized some commonly used conditional text datasets. Another key issue faced by researchers is how to evaluate the

<sup>25</sup><http://coai.cs.tsinghua.edu.cn/hml/dataset/#commonsense>

<sup>26</sup><https://conceptnet.io>

<sup>27</sup><https://visualdialog.org/data>

performance of these models and make fair and meaningful comparisons among them. Only with the help of reasonable evaluation metrics, can researchers accurately evaluate the performance of the designed models, to draw correct conclusions and promote applications of these models in real life. Natural language generation technology has made great progress in recent years, but the reasonable evaluation of the generated text is still a challenging problem to be solved. At present, researchers have not formed a unified theory on how to effectively evaluate text generation systems [125], and the lack of reasonable quantitative evaluation metrics will prevent the development of this field. There are mainly two kinds of evaluation metrics at the present stage, namely automated evaluation metrics and human evaluation metrics, which will be summarized below.

**4.11.1 Automatic Evaluation Metrics.** Automated machine evaluation is an intuitive and convenient method to evaluate the quality of text generation systems. The quality of the generated text is determined by uniquely designed formulas that compare the difference between the generated text and the ground-truth text. Among various evaluation metrics of text generation, the  $n$ -gram based metrics are the most widely studied which calculate the word overlap under the  $n$ -gram language unit between the generated text and ground-truth text. They are first applied in machine translation systems by calculating the degree of word overlap between the translated text and the target human-written references and then are widely used in the evaluation of many text generation systems. Typical evaluation metrics based on  $n$ -gram are summarized below.

**BLEU.** BLEU [102] is the harmonic mean of  $n$ -gram precisions of the generated texts with respect to ground-truth reference sentences, where  $n \in \{1, 2, 3, 4\}$ . The  $n$ -gram precisions refer to the proportion of the generated text that matches any  $n$ -gram unit in the reference sentences. Repeated  $n$ -gram matches are clipped to the maximum number of times the  $n$ -gram occurs in any single reference. BLEU contains many variants, such as SentBLEU [80],  $\Delta$  BLEU [41], NIST [33], and so on. NIST is a typical variant of BLEU, which improves BLEU's evaluation accuracy by assigning higher weights to low-frequency  $n$ -gram (e.g., more informative) and imposing length penalties.

**ROUGE-L.** Rouge-L [79] is calculated based on the length of the longest common subsequences (LCS) between the generated texts and the target texts, where the common subsequence needs to include the same words in the same order. Then the  $F$ -measure is calculated based on the maximum precision and recall of reference texts to obtain the final ROUGE-L score, where the accuracy and recall are calculated by dividing the length of LCS by the length of the generated text and the reference text respectively.

**METEOR.** METEOR [8] calculates the precision and recall of unigrams between generated texts and reference texts. In addition to the exact word matching, fuzzy matching is adopted in the calculation process based on the stem analysis and WordNet synonym. The matching degree is calculated with multiple reference texts and the best-matching one is selected as the final METEOR score.

**CIDEr.** CIDEr [127] is an evaluation metric designed for the image captioning task and can also be used for other text generation tasks. The CIDEr score is calculated by the average cosine similarity between the generated texts and the reference texts on the level of  $n$ -grams, where the importance of individual  $n$ -grams is calculated by the Term Frequency Inverse Document Frequency (TF-IDF) measure.

The above evaluation metrics based on  $n$ -gram have been widely used in various text generation tasks, including machine translation, dialogue system, text summarization, etc. However, numerous studies have shown that the  $n$ -gram matching cannot capture semantic information and effectively evaluate text generation models. In some specific applications with little variation, such as machine translation and question answering, metrics based on word-overlap of  $n$ -grams have a higher

evaluation accuracy. However, when reference texts have high diversity, such as dialogue systems and text summarization systems, it is difficult to make effective evaluation using these metrics.

In addition to evaluating the relationship between the generated text and the reference text, the characteristics of the natural language itself can also be used to evaluate the quality of text generation systems.

**Perplexity.** Perplexity [57] is a metric used to evaluate the quality of language models, which measures the average number of uncertain words when predicting words. The smaller the number is, the better the language model performance is. The fluency and diversity of generated text can be evaluated based on perplexity.

**4.11.2 Human Evaluation Metrics.** Most of the automatic evaluation metrics can only measure the quality of text generation models based on the similarity degree between generated texts and reference texts, but they cannot reflect the correctness, informativeness, naturalness, and other internal characteristics of the generated content. Therefore, human intelligence is introduced into the evaluation of text generation systems to provide more reasonable and effective evaluation. Human evaluation takes the form of the Turing test, which makes humans determine whether the quality of the machine-generated text is high enough to distinguish it from real data.

Human evaluation is done either on the overall quality of the generated text or at the finer-grained level, such as fluency, naturalness, informativeness, persona consistency, etc. Because the conditions considered in CTG systems (such as topics, knowledge, and personalized features) are difficult to evaluate using automatic evaluation metrics, human evaluation is almost the only and most effective way to evaluate CTG systems at the current research stage. For example, in knowledge-enhanced text generation systems, informativeness is a metric to measure whether the system effectively combines external knowledge. Only when external knowledge is correctly and reasonably integrated into the generation process, can more information-rich texts be generated. In personalized text generation systems, the persona consistency metric is used to evaluate whether the generated text conforms to the persona characteristics assigned to the text generation agent.

Human evaluation is the most effective gold standard for evaluating CTG systems. However, due to the subjectivity of human, there is often a high degree of variation in the process of human evaluation [125]. In addition, human evaluation is usually expensive, which requires a large amount of manpower to implement the evaluation process, and is not repeatable and costly. How to design reasonable evaluation metrics to make the evaluation of CTG systems more reasonable is still a challenging problem.

## 5 GENERAL LEARNING MODELS FOR CTG

We make a detailed investigation of various CTG fields under different conditions in above sections, which shows that combining different conditions can make the generated text more appropriate, informative and anthropomorphic. In this section, we attempt to address the CTG by distilling and presenting three different schemas of conditional generation models, including explicit conditional modeling, implicit conditional modeling and conditional knowledge transferring.

The explicit conditional modeling directly regards external conditions as a part of input, allowing the generator to process the condition information in the same way as the input information. This method is simple and efficient, which makes CTG systems easy to fuse the information of conditions from a global perspective, so as to generate results that are more consistent with external conditions. The implicit conditional modeling does not directly take conditions as input, but utilize specific algorithms, such as attention mechanism and RL, in the word generation stage to mine the implicit and deep semantic information in conditions, so as to make CTG systems more sensitive to conditions. Considering that the lack of conditional data makes it hard to fully train deep learning

models, the conditional knowledge transferring extracts general text knowledge from a large amount of text data and transfers it to CTG systems to improve the performance of CTG systems. The three general learning models mainly focus on how to fuse conditional information into the language generation process. Meanwhile, different decoding strategies can be applied to CTG systems to further control the appearance of certain conditions. Three general learning models for CTG are shown in figure 5.

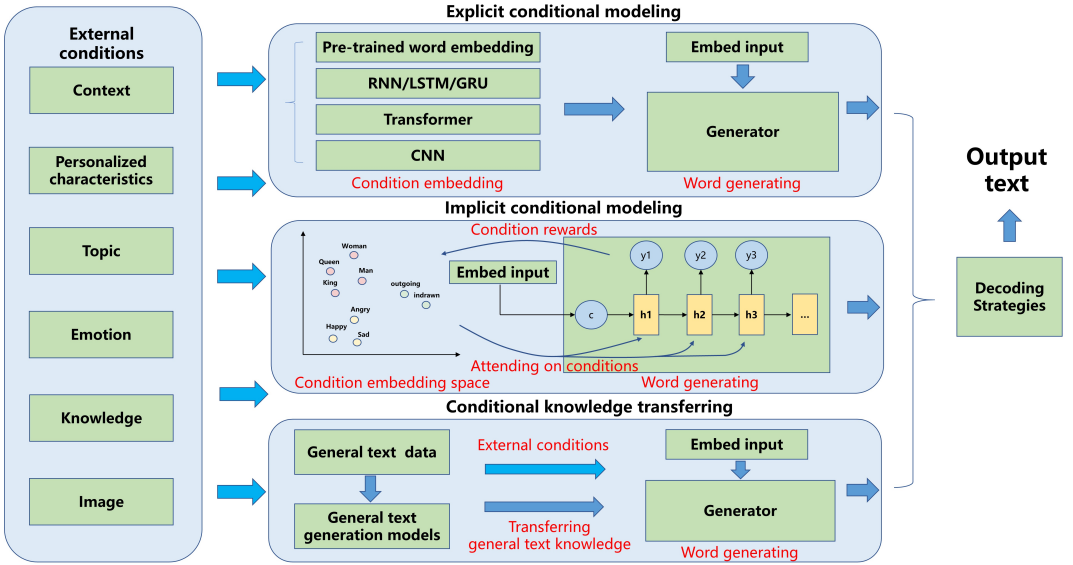


Fig. 5. General learning models for CTG

### 5.1 Explicit conditional modeling

In CTG systems, different conditions are in the form of natural language text (except for the image in the visual text generation), so taking them explicitly as a part of the input of the text generation system to enhance the input information is a straightforward method, known as explicit conditional modeling. In the same way that text generation systems process input, the condition will be transformed into vector representations by word embedding methods when the condition is a single word, or encoded by an additional neural networks after word embedding when the condition is a text sequence. Then vector representations of conditions will be used as the additional input information in the decoding stage to guide the text generation. Explicit conditional modeling is a simple, straightforward and proven method to integrate external conditions in CTG systems, without increasing the complexity of the model.

For example, in context-based text generation systems, Sordoni *et al.* [121] regard the dialogue history as the context information and embed all words and phrases in the dialogue history into vector representations which are then decoded by another RNN to promote context-aware responses. In personalized text generation systems, Li *et al.* [71] propose persona-based models to solve the problem of speaker consistency in dialogue systems. The *Speaker Model* maps each individual speaker into a vector by encoding speaker-specific information, such as age, gender, and dialect, that may influence his/her speaking content and style. Then speaker vectors are injected into decoder hidden layers to generate personalized dialogue responses. For knowledge-enhanced text

generation systems, Ghazvininejad *et al.* [45] propose a knowledge-grounded dialogue model to produce contentful and informative dialogue responses. Given the dialogue history, the relevant knowledge facts are extracted from knowledge base using words in the dialogue history as keywords. Then the *Facts Encoder* module adopts Memory Network to encode facts into vector representations based the user input and dialogue history. This module enables the dialogue system to deeper exploit inter lexical dependencies between different parts of facts and the input for effectively fusing knowledge into generation process. Then both the encoded dialogue history and knowledge facts are fed into the decoder to generate context relevant and informative responses.

## 5.2 Implicit conditional modeling

In order to capture the deeper semantic information contained in the condition, the condition information is not directly fed into the decoder after embedding, but interact with the decoding state at a deeper level using additional algorithms, known as implicit conditional modeling. The attention mechanism is a typical implicit condition modeling method, which can dynamically capture the important part of the condition according to decoder states, so as to realize implicit condition modeling and make the condition information guide the text generation process more effectively. The RL mechanism can give different rewards to actions of different states so as to give feedback on whether specific conditions are reflected in the generated results. Therefore, RL is suitable for implicit conditional modeling. By penalizing results that do not reflect conditional information, the RL mechanism forces the model to consider more conditional information in generation.

For example, Zheng *et al.* [159] propose the persona aware attention mechanism to control the attention weights of context vectors under integrated persona vectors and the persona-aware bias to estimate the generation distribution for personalized dialogue responses generation. Zhou *et al.* [163] produce two graph attention mechanisms, *static graph attention* and *dynamic graph attention* respectively, to promote dialogue understanding and knowledgeable responses generating. The model extracts relevant knowledge graphs using the entities in the input as keywords and then encode graphs into static vector representations by the *static graph attention* module, which considers all nodes and relations between nodes in a graph to encode more structural semantic information. In the decoder stage, the *dynamic graph attention* attends the knowledge graphs and knowledge triples in each graph to efficiently integrate information in knowledge graphs for informative response generation.

Li *et al.* [74] propose a RL-based dialogue model combined with an emotional editor module to generate customizable emotional responses. The emotional editor selects the template sentence according to the given topic and emotion to provide references for the generation process. The RL mechanism constrains the quality of generated responses from three points: emotion, topic and coherence, to generate emotional, topic-relevant and meaningful dialogue responses. The multi-task learning is also introduced to enhance the model discrimination to learn the coherence, topic, and emotion of a reply.

## 5.3 Conditional knowledge transferring

Compared with general text generation, CTG lacks available conditional text datasets. Although researchers have released several conditional text datasets, it is generally not enough to train a well-performing CTG system. By extracting general text knowledge from a large amount of text data and transferring it to CTG systems, well-performing CTG systems can be trained in the absence of conditional data, known as conditional knowledge transferring.

For personalized text generation, Luan *et al.* [84] train a dialogue model to predict responses given previous contexts and an autoencoder model with large volumes of non-conversational personal



data to model the role-specific characteristics of different users. Through the multi-task learning mechanism which shares the decoder parameters of the two models, these models can capture speaker roles, expressive styles and domain expertise characteristic of the targeted user and generate personalized responses without heavy recourse to each speaker's conversational data. The work of Yang *et al.* [145] utilizes domain adaptation mechanism to generate personalized dialogue responses. The general response generation model with an attention LSTM encoder-decoder architecture is pre-trained on the large-scale general dialogue data without user-specific information. Then the model is fine-tuned with a small amount of personalized dialogue data by a dual learning mechanism to generate personalized responses. In the training process, three rewards are proposed to evaluate the quality of generation results, that are personalization, informativeness, and grammaticality and the policy gradient method is adopted to generate highly rewarded responses.

## 6 OPEN ISSUES AND FUTURE TRENDS

Although many advanced technologies have been applied to text generation and some remarkable achievements have been made, there are still many remaining issues. In this section, we put forward some key issues and point out some future development trends of text generation.

### 6.1 Different Types of Contexts

Context information can provide the current situation, state, environment and other information for text generation systems to realize more accurate simulation of human expression, so it is very important in CTG systems. For example, in multi-turn dialogue systems, context information usually refers to historical dialogues and can make the conversation more coherent. Sordoni *et al.* [121] encode the dialogue history into vector representations and feed them into the decoder to generate consistent responses. In machine translation systems, Voita *et al.* [131] propose a context-aware machine translation model, which can control and analyze the flow of information from context to the translation model. By encoding the context information, the model can more accurately generate the correct translation text. These works achieve relatively excellent results. However, context information contains much more than those considered in existing studies. Existing researches only consider the dialogue history, predefined external information or other types context, but our expression may be affected by various factors, such as hot events, emotions, time, weather, etc. These external context information may be explicit or implicit, so how to extract them and properly represent them is a major challenge. A humanoid CTG system should be able to effectively integrate various contexts and generate reasonable text, which will be the future research direction.

### 6.2 Multi-modal Data Translation and Domain Adaptation

In addition to text data, there are various types of data, such as voice, image and video. Human can efficiently extract useful information from various types of data and convert them into corresponding text representation, such as describing the content of a painting and summarizing the content of a movie. Researchers carry out many works on text generation with multi-modal data as inputs, such as generating description/caption for a given image [83], conducting QA with images [119], and communicating based on the content of a given image [28]. These studies usually utilize CNN to extract relevant information in images, and generate corresponding text using common models in text generation. The existing deep learning models can only process one type of data, so handling multi-modal data requires to combine multiple models, which may bring a large computational cost. Moreover, the semantic space of different types of data may be different, which brings great difficulties for the fusion of multi-modal data. How to incorporate different types of data and develop unified models for multi-source data processing are two huge challenges. Much more efforts should be conducted on generating informative texts with multi-modal data sources.

At the same time, in many tasks of CTG, such as personalized text generation and emotional text generation, the available training data is very scarce. Most text data are totally general data and do not contain personalized or emotional characteristics, which cannot meet the requirements under specific conditions. Transfer learning is a promising way to address this problem. By learning general knowledge of natural language from massive common text data, and then transferring it to a specific domain training with a small-scale conditional text data, the model can not only master the general knowledge of the source domain, but also learn the specific needs of the target domain, to make up for the scarcity of data. Yang *et al.* [145] use the idea of domain adaptation in transfer learning to address the issue of lacking personalized dialogue data. Through fine-tuning the general dialogue model with the small size personalized dialogue data, the model can effectively generate personalized dialogue responses. Transfer learning is a rapidly developing technology in deep learning, and integrating diverse transfer learning models with scarce usable data for CTG is also a promising research direction.

### 6.3 Long Text Generation

Long text has a wide range of application areas, including writing compositions, translating articles, writing reports, etc. However, the current technology has some bottlenecks in processing long text because of the long-distance dependence existing in the natural language. Humans have the ability to extract the key information (*e.g.*, contexts and topics) from long text, which is difficult for machines. Researchers have conducted much efforts on improving the models' ability of generating long text. For example, the LSTM and GRU model is produced to address the issue that the original RNN model cannot capture the long-distance dependence using the gating mechanism. Guo *et al.* [50] propose the *LeakGAN* model to generate long texts. The feature extracted by discriminator is used as a stepwise guidance signal to guide the generator to produce high-quality text. At the same time, the hierarchical reinforcement learning provides more information to the discriminator for generating long text. How to effectively model and capture the long-term dependency in long text is a major challenge in the research. For text generation technology truly behaving like humans, it needs the ability to freely generate long or short texts, which still has much to be investigated.

### 6.4 Lifelong learning

Lifelong learning is an important ability of human beings. We continuously learn new knowledge, expand and update our knowledge base through various data sources in the physical world to adapt to the rapidly changing environment. Existing text generation models are usually trained on fixed datasets and have no ability to expand and dynamically updated according to the changes of the external environment. In order to make text generation models more personified, lifelong learning is a necessary ability. Combining external knowledge base is the first step to realize lifelong learning. There have been many text generation researches, focusing on the dialogue systems [163] [32], combined with knowledge bases. However, most of them are based on fixed knowledge base in which the knowledge does not keep updating in real time, so the model still does not have the ability of continuous learning. Therefore, the dynamic evolution of knowledge base is very important. Mazumder *et al.* [93] propose a lifelong learning model which can update its own knowledge base by asking users. This is a groundbreaking exploration which provides a direction for the lifelong learning text generation models. However, the ability to actively gain knowledge from the environment rather than simply asking the user is very important. How to find and learn the most effective information from the changeable external environment and achieve efficient lifelong learning is a very important research direction in CTG.

### 6.5 Knowledge extraction and fusion from crowdsourced data

With the rapid development and popularization of social networks, more and more crowdsourced data appear on the Web, such as the Q&A community Quora<sup>28</sup> and Zhihu<sup>29</sup>. These data are the embodiment of human intelligence and can be used as the source of external knowledge of text generation systems. However, existing knowledge-enhanced text generation systems are mainly based on the structured knowledge graph or unstructured knowledge base built in advance [78][98], which cannot perform real-time knowledge selection and fusion from crowdsourced data and usually cover several specific domains. Crowd intelligence data covers a wide range of domains and dynamically updates itself in real time. However, the noise and heterogeneity of crowdsourced data prevent its applications in text generation systems. Therefore, how to mine suitable information from crowdsourced intelligence data in real time, enhance the semantic understanding, and fuse information properly in text generation, are promising research directions of CTG.

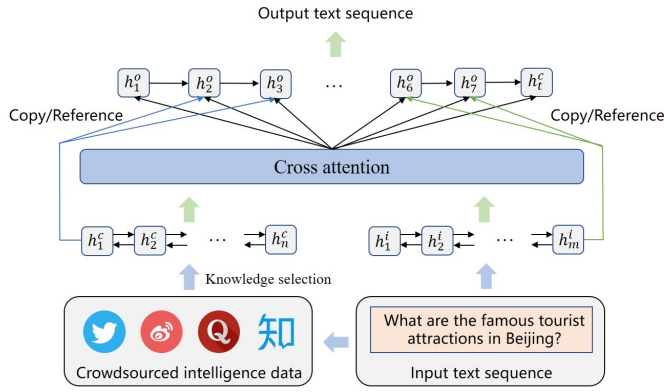


Fig. 6. The concept map of our proposed model

To solve the problem of incorporating crowdsourced intelligence data in CTG, we have proposed a preliminary idea, as shown in figure 6. In the encoding stage, the input text will interact with the crowdsourced data and select relevant knowledge as the external knowledge source. Then the knowledge and the input text are encoded separately, and the cross-attention mechanism is used to obtain fusion vector representations of the input information. In the decoding stage, generation and copy/reference mechanisms are considered at the same time. The decoder will dynamically select whether to generate words from the vocabulary or copy words from input sources according to the current decoding state, to achieve more explicit use of crowdsourced knowledge.

## 7 CONCLUSION

We have made a systematic review of the research trends of conditional text generation (CTG). In this paper, we first give an introduction to the field of text generation. We then give a brief review of key techniques in the field of text generation and further give the formal definitions of different fields of CTG. Finally, we investigate the research status of various CTG fields and propose several general learning models for CTG. Though there has been a big research progress in CTG, it is still at the early stage and numerous open research issues and promising research directions should be studied, such as long text generation, multimodal data translation, and lifelong learning.

<sup>28</sup><https://www.quora.com/>

<sup>29</sup><https://www.zhihu.com/>

## ACKNOWLEDGMENTS

This work was partially supported by the The National Science Fund for Distinguished Young Scholars (62025205), and the National Natural Science Foundation of China (No. 61772428, 61725205)

## REFERENCES

- [1] Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. 2019. Sequential latent spaces for modeling the intention during diverse image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*. 4261–4270.
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 2425–2433.
- [3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*. 214–223.
- [4] Nabihha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. 2018. Affective neural response generation. In *European Conference on Information Retrieval*. Springer, 154–166.
- [5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- [6] Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating More Interesting Responses in Neural Conversation Models with Distributional Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3970–3980.
- [7] Peng Bai, Yang Xia, and Yongsheng Xia. 2020. Fusing Knowledge and Aspect Sentiment for Explainable Recommendation. *IEEE Access* 8 (2020), 137150–137160.
- [8] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.
- [9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
- [10] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 10–21.
- [11] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
- [12] Ginevra Carbone and Gabriele Sarti. 2020. ETC-NLG: End-to-end Topic-Conditioned Natural Language Generation. *arXiv preprint arXiv:2008.10875* (2020).
- [13] Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. CoCon: A Self-Supervised Approach for Controlled Text Generation. *arXiv preprint arXiv:2006.03535* (2020).
- [14] Zhangming Chan, Juntao Li, Xiaopeng Yang, Xiuying Chen, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Modeling personalization in continuous space for response generation via augmented wasserstein autoencoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1931–1940.
- [15] Fuhai Chen, Rongrong Ji, Jiayi Ji, Xiaoshuai Sun, Baochang Zhang, Xuri Ge, Yongjian Wu, Feiyue Huang, and Yan Wang. 2019. Variational Structured Semantic Inference for Diverse Image Captioning. In *Advances in Neural Information Processing Systems*. 1931–1941.
- [16] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter* 19, 2 (2017), 25–35.
- [17] Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, and Zhipeng Guo. 2019. Sentiment-controllable Chinese poetry generation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 4925–4931.
- [18] Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Towards Knowledge-Based Personalized Product Description Generation in E-commerce. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 3040–3050.
- [19] Shuang Chen, Jinpeng Wang, Xiaocheng Feng, Feng Jiang, Bing Qin, and Chin-Yew Lin. 2019. Enhancing Neural Data-To-Text Generation Models with External Background Knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3013–3023.

- [20] Sheng Yeh Chen, Chao Chun Hsu, Chuan Chun Kuo, Kenneth Huang, and Lun Wei Ku. 2019. Emotionlines: An emotion corpus of multi-party conversations. In *11th International Conference on Language Resources and Evaluation, LREC 2018*. European Language Resources Association (ELRA), 1597–1601.
- [21] Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling Knowledge Learned in BERT for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7893–7905.
- [22] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.
- [23] Sajal Choudhary, Prerna Srivastava, Lyle Ungar, and João Sedoc. 2017. Domain aware neural dialog system. *arXiv preprint arXiv:1708.00897* (2017).
- [24] Elizabeth Clark, Yangfeng Ji, and Noah A Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2250–2260.
- [25] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).
- [26] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*. 2970–2979.
- [27] Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd workshop on cognitive modeling and computational linguistics*. Association for Computational Linguistics, 76–87.
- [28] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 326–335.
- [29] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*.
- [30] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of NAACL-HLT*. 2306–2317.
- [31] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019. Learning to Transform, Combine, and Reason in Open-Domain Question Answering. In *WSDM*. 681–689.
- [32] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.
- [33] George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*. 138–145.
- [34] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*. 13063–13075.
- [35] Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar R Zaiane. 2019. Augmenting Neural Response Generation with Context-Aware Topical Attention. In *Proceedings of the First Workshop on NLP for Conversational AI*. 18–31.
- [36] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.
- [37] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 889–898.
- [38] Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-Essay Generation with Neural Networks. In *IJCAI*. 4078–4084.
- [39] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. 2019. Unsupervised image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4125–4134.
- [40] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [41] Michel Galley, Chris Brockett, Alessandro Sordani, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 445–450.
- [42] Ce Gao and Jiangtao Ren. 2019. A topic-driven language model for learning to generate diverse sentences. *Neurocomputing* 333 (2019), 374–380.
- [43] Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational AI. *Foundations and Trends® in Information Retrieval* 13, 2-3 (2019), 127–298.

- [44] Pierre-Etienne Genest and Guy Lapalme. 2011. Framework for abstractive summarization using text-to-text generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Association for Computational Linguistics, 64–73.
- [45] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [46] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*. 43–48.
- [47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
- [48] Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and common-sense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6473–6480.
- [49] Beliz Gunel, Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2019. Mind The Facts: Knowledge-Boosted Coherent Abstractive Text Summarization. In *NeurIPS 2019*.
- [50] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [51] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*. 1693–1701.
- [52] Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*. 252–256.
- [53] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.
- [54] Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1638–1649.
- [55] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1587–1596.
- [56] Aaron Jaech and Mari Ostendorf. 2018. Low-rank RNN adaptation for context-aware language modeling. *Transactions of the Association for Computational Linguistics* 6 (2018), 497–510.
- [57] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America* 62, S1 (1977), S63–S63.
- [58] Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic Context Selection for Document-level Neural Machine Translation via Reinforcement Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2242–2254.
- [59] Chien-Hao Kao, Chih-Chieh Chen, and Yu-Tza Tsai. 2019. Model of Multi-turn Dialogue in Emotional Chatbot. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, 1–5.
- [60] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.
- [61] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).
- [62] Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=Hke0K1HKwr>
- [63] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [64] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [65] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of NAACL-HLT*. 2284–2293.
- [66] Xiang Kong, Bohan Li, Graham Neubig, Eduard Hovy, and Yiming Yang. 2019. An Adversarial Approach to High-Quality, Sentiment-Controlled Neural Dialogue Generation. *arXiv preprint arXiv:1901.07129* (2019).
- [67] Kundan Krishna and Balaji Vasani Srinivasan. 2018. Generating topic-oriented summaries using neural attention. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1697–1705.
- [68] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

- [69] Chunyuan Li, Xiang Gao, Yuan Li, Xiujuan Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092* (2020).
- [70] Hui Li, Peng Wang, Chunhua Shen, and Anton van den Hengel. 2019. Visual Question Answering as Reading Comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6319–6328.
- [71] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 994–1003.
- [72] Junjie Li, Haoran Li, and Chengqing Zong. 2019. Towards Personalized Review Summarization via User-Aware Sequence Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6690–6697.
- [73] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562* (2016).
- [74] Jia Li, Xiao Sun, Xing Wei, Changliang Li, and Jianhua Tao. 2019. Reinforcement Learning Based Emotional Editing Constraint Conversation Generation. *arXiv preprint arXiv:1904.08061* (2019).
- [75] Miao Li, Beihong Jin, et al. 2019. A Topic Augmented Text Generation Model: Joint Learning of Semantics and Structural Features. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5093–5102.
- [76] Margaret Li, Stephen Roller, Ilya Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2019. Don't Say That! Making Inconsistent Dialogue Unlikely with Unlikelihood Training. *arXiv preprint arXiv:1911.03860* (2019).
- [77] Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019. Persona-Aware Tips Generation?. In *The World Wide Web Conference*. 1006–1016.
- [78] Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental Transformer with Deliberation Decoder for Document Grounded Conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 12–21.
- [79] Chin-Yew Lin and FJ Och. 2004. Looking for a few good metrics: ROUGE and its evaluation. In *Ntcir Workshop*.
- [80] Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. 501–507.
- [81] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [82] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909* (2015).
- [83] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 375–383.
- [84] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-Task Learning for Speaker-Role Adaptation in Neural Conversation Models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 605–614.
- [85] Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2013–2022.
- [86] Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. 2019. Learning personalized end-to-end goal-oriented dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6794–6801.
- [87] Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020. The Adapter-Bot: All-In-One Controllable Conversational Model. *arXiv preprint arXiv:2008.12579* (2020).
- [88] Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking Emotions for Empathetic Response Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8968–8979.
- [89] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE international conference on computer vision*. 1–9.
- [90] Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison Cottrell. 2019. Improving Neural Story Generation by Targeted Common Sense Grounding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5990–5995.
- [91] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090* (2014).

- [92] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *EMNLP*.
- [93] Sahisnu Mazumder, Nianzu Ma, and Bing Liu. 2018. Towards a continuous knowledge learning engine for chatbots. *arXiv preprint arXiv:1802.06024* (2018).
- [94] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. 6294–6305.
- [95] Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.
- [96] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [97] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 845–854.
- [98] Diego Moussallem, Mihael Arčan, Axel-Cyrille Ngonga Ngomo, and Paul Buitelaar. 2019. Augmenting neural machine translation with knowledge graphs. *arXiv preprint arXiv:1902.08816* (2019).
- [99] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *CoNLL 2016* (2016), 280.
- [100] Jianmo Ni and Julian McAuley. 2018. Personalized Review Generation by Expanding Phrases and Attending on Aspect-Aware Representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 706–711.
- [101] Will Oremus. 2014. The first news report on the LA earthquake was written by a robot. *Slate.com* 17 (2014).
- [102] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.
- [103] Yehong Peng, Yizhen Fang, Zhiwen Xie, and Guangyou Zhou. 2019. Topic-enhanced emotional conversation generation with attention mechanism. *Knowledge-Based Systems* 163 (2019), 429–437.
- [104] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*. 2227–2237.
- [105] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational linguistics* 28, 4 (2002), 399–408.
- [106] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL [https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language\\_understanding\\_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf) (2018).
- [107] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
- [108] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).
- [109] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 5370–5381.
- [110] Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2019. Thinking Globally, Acting Locally: Distantly Supervised Global-to-Local Knowledge Selection for Background Based Conversation. *arXiv preprint arXiv:1908.09528* (2019).
- [111] Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2017. A Neural Attention Model for Sentence Summarization. In *ACLWeb. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
- [112] Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? How controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1702–1723.
- [113] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- [114] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [115] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1577–1586.



- [116] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, et al. 2019. Long and Diverse Text Generation with Planning-based Hierarchical Variational Model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3248–3259.
- [117] Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. Towards Generating Long and Coherent Text with Multi-Level Latent Variable Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2079–2089.
- [118] Haoyu Song, Yan Wang, Wei-Nan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, Delete and Rewrite: A Three-Stage Framework for Improving Persona Consistency of Dialogue Generation. *arXiv preprint arXiv:2004.07672* (2020).
- [119] Jingkuan Song, Pengpeng Zeng, Lianli Gao, and Heng Tao Shen. 2018. From Pixels to Objects: Cubic Visual Attention for Visual Question Answering. In *IJCAI*. 906–912.
- [120] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *International Conference on Machine Learning*. 5926–5936.
- [121] Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 196–205.
- [122] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
- [123] Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *arXiv preprint arXiv:1611.09900* (2016).
- [124] Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 231–236.
- [125] Chris Van Der Lee, Albert Gatt, Emiel Van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*. 355–368.
- [126] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [127] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4566–4575.
- [128] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).
- [129] Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869* (2015).
- [130] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.
- [131] Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1264–1274.
- [132] Ke Wang and Xiaojun Wan. 2018. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. In *IJCAI*. 4446–4452.
- [133] Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization. In *International Joint Conference on Artificial Intelligence*.
- [134] Run-Ze Wang, Zhen-Hua Ling, and Yu Hu. 2019. Knowledge base question answering with attentive pooling for question representation. *IEEE Access* 7 (2019), 46773–46784.
- [135] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-Guided Variational Auto-Encoder for Text Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 166–177.
- [136] Yanmeng Wang, Wenge Rong, Yuanxin Ouyang, and Zhang Xiong. 2019. Augmenting Dialogue Response Generation With Unstructured Textual Knowledge. *IEEE Access* 7 (2019), 34954–34963.
- [137] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural Text Generation With Unlikelihood Training. In *International Conference on Learning Representations*.
- [138] Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association*

for *Computational Linguistics*. 5811–5820.

- [139] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [140] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [141] Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [142] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.
- [143] An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2346–2357.
- [144] Min Yang, Qiang Qu, Kai Lei, Jia Zhu, Zhou Zhao, Xiaojun Chen, and Joshua Z Huang. 2018. Investigating deep reinforcement learning techniques in personalized dialogue generation. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 630–638.
- [145] Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. 2017. Personalized response generation via domain adaptation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1021–1024.
- [146] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. 5754–5764.
- [147] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [148] 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*.
- [149] Chenhan Yuan and Yi-Chin Huang. 2019. Personalized sentence generation using generative adversarial networks with author-specific word usage. *arXiv preprint arXiv:1904.09442* (2019).
- [150] Haisong Zhang, Zhangming Chan, Yan Song, Dongyan Zhao, and Rui Yan. 2018. When Less Is More: Using Less Context Information to Generate Better Utterances in Group Conversations. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 76–84.
- [151] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2204–2213.
- [152] Wei-Nan Zhang, Qingfu Zhu, Yifa Wang, Yanyan Zhao, and Ting Liu. 2019. Neural personalized response generation as domain adaptation. *World Wide Web* 22, 4 (2019), 1427–1446.
- [153] Yizhe Zhang, Zhe Gan, and Lawrence Carin. 2016. Generating text via adversarial training. In *NIPS workshop on Adversarial Training*, Vol. 21.
- [154] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536* (2019).
- [155] Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: Constrained Text Generation via Insertion-based Generative Pre-training. *arXiv preprint arXiv:2005.00558* (2020).
- [156] Xueliang Zhao, Chongyang Tao, Wei Wu, Can Xu, Dongyan Zhao, and Rui Yan. 2019. A document-grounded matching network for response selection in retrieval-based chatbots. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 5443–5449.
- [157] Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020. Low-Resource Knowledge-Grounded Dialogue Generation. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=rJelcTNTvS>
- [158] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-Grounded Dialogue Generation with Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3377–3390.
- [159] Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672* (2019).
- [160] Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*

- the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 165–176.
- [161] Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards Persona-Based Empathetic Conversational Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6556–6566.
  - [162] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
  - [163] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention.. In *IJCAI*. 4623–4629.
  - [164] Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A Dataset for Document Grounded Conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 708–713.
  - [165] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. [n.d.]. The design and implementation of XiaoIce, an empathetic social chatbot. *Computational Linguistics* Just Accepted ([n. d.]), 1–62.
  - [166] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4995–5004.
  - [167] Zachary M Ziegler, Luke Melas-Kyriazi, Sebastian Gehrmann, and Alexander M Rush. 2019. Encoder-agnostic adaptation for conditional language generation. *arXiv preprint arXiv:1908.06938* (2019).