A Survey of Natural Language Generation

CHENHE DONG[*], Sun Yat-Sen University, China
YINGHUI LI[*], Tsinghua University, China
HAIFAN GONG, Sun Yat-Sen University, China
MIAOXIN CHEN, Tsinghua University, China
JUNXIN LI, Tsinghua University, China
YING SHEN[†], Sun Yat-Sen University, China
MIN YANG[†], Chinese Academy of Science, China

This paper offers a comprehensive review of the research on Natural Language Generation (NLG) over the past two decades, especially in relation to data-to-text generation and text-to-text generation deep learning methods, as well as new applications of NLG technology. This survey aims to (a) give the latest synthesis of deep learning research on the NLG core tasks, as well as the architectures adopted in the field; (b) detail meticulously and comprehensively various NLG tasks and datasets, and draw attention to the challenges in NLG evaluation, focusing on different evaluation methods and their relationships; (c) highlight some future emphasis and relatively recent research issues that arise due to the increasing synergy between NLG and other artificial intelligence areas, such as computer vision, text and computational creativity.

CCS Concepts: • **Computing methodologies → Natural language generation**.

Additional Key Words and Phrases: natural language generation, data-to-text generation, text-to-text generation, deep learning, evaluation

ACM Reference Format:
Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2021. A Survey of Natural Language Generation. *ACM Comput. Surv.* 1, 1, Article 1 (January 2021), 36 pages. https://doi.org/xxx.xxx

1 INTRODUCTION

This paper surveys the current state of the art in Natural Language Generation (NLG), defined as the task of generating text from underlying non-linguistic representation of information [85]. NLG has been receiving more and more attention from researchers because of its extremely challenging and promising application prospects.

[*] Equal contribution
[†] Corresponding author

Authors' addresses: Chenhe Dong, dongchh@mail2.sysu.edu.cn, Sun Yat-Sen University, Guangzhou, China, 510275; Yinghui Li, liyinghu20@mails.tsinghua.edu.cn, Tsinghua University, Shenzhen, China, 510100; Haifan Gong, haifangong@outlook.com, Sun Yat-Sen University, Guangzhou, China, 510275; Miaoxin Chen, cmx20@mails.tsinghua.edu.cn, Tsinghua University, Shenzhen, China, 510100; Junxin Li, ljx20@mails.tsinghua.edu.cn, Tsinghua University, Shenzhen, China, 510100; Ying Shen, sheny76@mail.sysu.edu.cn, Sun Yat-Sen University, Guangzhou, China, 510275; Min Yang, min.yang@siat.ac.cn, Chinese Academy of Science, Shenzhen, China, 510100.

0360-0300/2021/1-ART1 $15.00
https://doi.org/xxx.xxx

1.1 What is Natural Language Generation?

Natural Language Generation (NLG) is the process of producing a natural language text in order to meet specified communicative goals. The texts that are generated may range from a single phrase given in answer to a question, through multi-sentence remarks and questions within a dialog, to full-page explanations.

In contrast with the organization of the Natural Language Understanding (NLU) process–which can follow the traditional stages of a linguistic analysis: morphology, syntax, semantics, pragmatics/discourse–the generation process has a fundamentally different character. Generation proceeds involve the content planning, determination and realization from content to form, from intentions and perspectives to linearly arrayed words and syntactic markers. Coupled with its application, the situation, and the discourse, they provide the basis for making choices among the alternative wordings and constructions that the language provides, which is the primary effort in constructing a text deliberately [62]. With its opposite flow of information, one might assume that a generation process could be organized like an understanding process but with the stages in opposite order.

Both data-to-text generation and text-to-text generation are instances of NLG. Generating text from image is an application of data-to-text generation. A further text-to-text generation complication is dividing NLG tasks into three categories, i.e., text abbreviation, text expansion, text rewriting and reasoning. The text abbreviation task is formulated to condense information from long texts to short ones, typically including research on text summarization [6, 7, 15, 17, 43, 80, 99], question generation [4, 18, 34, 36, 53, 95, 104, 112, 113, 130, 134], and distractor generation [22, 50, 60, 73, 82, 86, 100, 101]. The text expansion tasks, such as short text expansion [5, 89, 96, 106] and topic-to-essay generation [19, 81, 114, 123, 129], generate complete sentences or even texts from some meaningful words by considering and adding elements like conjunctions and prepositions to transform the input words into linguistically correct outputs. The goal of text rewriting and reasoning task is to rewrite the text into another style or applying reasoning methods to create responses. There are two sub-tasks: text style transfer [9, 20, 33, 58, 64, 69, 79, 120, 133], and dialogue generation [3, 35, 46, 49, 59, 115, 122, 135]. The task of visual based text generation targets at generate the explanation or summarization of the given image or video, involving the study of image caption [1, 57, 87, 110, 125, 126], video caption [16, 39, 42, 72, 97, 103, 109, 111, 118, 119, 136, 137], and visual storytelling [30, 45, 124].

1.2 Why a Survey on Natural Language Generation?

Here, we will explain the reasons and motivations why the natural language generation is worth reviewing and investigating.

Reiter *et al.* [85] provided the most classical survey of NLG. However, the field of NLG has changed drastically in the last 20 years, with the emergence of successful deep learning methods. For example, since 2014, various neural encoder-decoder models pioneered by Seq2Seq have been proposed to achieve the goal by learning to map input text to output text. In addition, the evaluation of NLG output should start to receive systematic attention.

Since Reiter *et al.* [85] published their book, various other NLG overview texts have also appeared. Gatter *et al.* [23] introduce the core tasks, applications and evaluation metrics of natural language generation. While useful, this survey is not highly timely and does not include the state-of-the-art research on the novel deep learning models such as graph neural networks. Perera *et al.* [76] cover some tasks or architectures of NLG. Santhanam *et al.* [91] review the NLG research progress of dialogue systems. Mogandala *et al.* [66] study the integration of vision and language in multimodal NLG, such as image dialogue and video storytelling. Otter *et al.* [70] concludes the progress of deep learning for NLP, display some classic text generation methods but barely discuss the research progress of NLG. Yu *et al.* [128] offers a survey of the knowledge-enhanced text generation methods.

The goal of the our survey is to present a highly timely overview of NLG developments from the aspect of data-to-text generation and text-to-text generation. Though NLG has been a part of AI and Natural language processing for long time, it has only recently begun to take full advantage of recent advances in data-driven, machine learning and deep learning approaches. Therefore this survey will focus on introducing the latest development and future directions of deep learning methods in field of NLG. This survey can have broad audiences, researchers and practitioners, in academia and industry.

1.3 Contribution of this Survey

In this paper, we provide a thorough review of different natural language generation tasks as well as its corresponding datasets and methods. To summarize, this paper presents an extensive survey of natural language generations with the following contributions:

(1) To give an up-to-date synthesis of deep learning research on the core tasks in NLG, as well as the architectures adopted in the field;
(2) To detail meticulously and comprehensively various NLG tasks and datasets, and draw attention to the challenges in NLG evaluation, focusing on different evaluation methods and their relationships.
(3) To highlight some future emphasis and relatively recent research issues that arise due to the increasing synergy between NLG and other artificial intelligence areas, such as computer vision, text and computational creativity.

The rest of this survey is organized as follows. In Sec.2, we introduce the general methods of NLG to give a comprehensive understanding. From Sec.3 to Sec.6, we will give a comprehensive introduction to the four main areas of NLG from the perspectives of task, data, and methods. In Sec.7, we present the important evaluation metrics used in various aforementioned NLG tasks. Besides, we propose some problems and challenges of NLG as well as several future research directions in Sec.8. And finally we conclude our survey in Sec.9.

2 GENERAL METHODS OF NLG

In general, the task of natural language generation (NLG) targets at finding an optimal sequence $y_{<T+1} = (y_1, y_2, ..., y_T)$ that satisfies:

$$\begin{aligned} y_{<T+1} &= \mathop{\arg\max}_{y_{<T+1} \in \mathcal{Y}} \log P_\theta(y_{<T+1}|x) \\ &= \mathop{\arg\max}_{y_{<T+1} \in \mathcal{Y}} \sum_{t=1}^{T} \log P_\theta(y_t|y_{<t}, x), \end{aligned} \tag{1}$$

where T represents the number of tokens of the generated sequence, $\mathcal{Y}$ represents a set containing all possible sequences, and $P_\theta(y_t|y_{<t}, x)$ is the conditional probability of the next token y_t based on its previous tokens $y_{<t} = (y_1, y_2, ..., y_{t-1})$ and the source sequence x with model parameters θ.

The general methods to deal with the tasks of NLG mainly contain: RNN Seq2Seq, Transformer, Attention Mechanism, Copy and Pointing Mechanisms, GAN, Memory Network, GNN, and Pretrained Model.

2.1 RNN Seq2Seq

As proposed by [105], the encoder of Seq2Seq is an RNN, it will traverse every token (word) of the input, the input of each time is the hidden state and input of the previous time, and then there will be an output and a new hidden state. The new hidden state will be used as the input hidden state of the next time. We usually only keep the hidden state of the last time, which encodes the semantics of the whole sentence. After the encoder processing, the last hidden state will be regarded as the initial hidden state of the decoder. The Decoder is also an RNN, which outputs one word at a time. The input of each time is the hidden state of the previous time and the output of the previous time. The initial hidden state is the last hidden state of the encoder, and the input is special. Then, RNN is used to calculate the new hidden state and output the first word, and then the new hidden state and the first word are used to calculate the second word. Until EOS is encountered and the output is finished. A standard RNN computes a sequence of outputs $(y_1, ..., y_T)$ given a sequence of inputs $(x_1, ..., x_T)$ by iterating the following equation:

$$h_t = \mathrm{sigm}(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}), \tag{2}$$

$$y_t = W^{\mathrm{yh}} h_t. \tag{3}$$

2.2 Transformer

Transformer [107] is based on the encoder-decoder framework, and both encoder and decoder are composed of stacked identified layers. The encoder is used to map an input sequence of symbol representations to another sequence of continuous representations, and then the decoder auto-regressively generates an output sequence based on its previously generated symbols and the continuous representations from encoder. In the encoder, each layer contains two sub-layers, which are multi-head self-attention mechanism (MultiHeadAttn) and position-wise fully connected feed-forward network (FFN) respectively. The multi-head self-attention can be formulated by:

$$\mathrm{MultiHeadAttn}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O, \tag{4}$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{5}$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^\top}{\sqrt{d_k}})V, \tag{6}$$

where Q, K, V are the query, key, and value matrices, d_k is the dimension of queries and keys. And the feed-forward network can be formulated by:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2. \tag{7}$$

Each of the stacked layer is surrounded by a residual connection followed by layer normalization. While in the decoder, each layer contains three sub-layers with an additional multi-head attention over the encoder's output. The self-attention in decoder is masked to prevent attending to subsequent tokens. In addition, to inject the position information, the position encodings are added to the input embeddings as formulated by:

$$PE_{pos,2i} = sin(pos/10000^{2i/d_{model}}), \tag{8}$$

$$PE_{pos,2i+1} = cos(pos/10000^{2i/d_{model}}), \tag{9}$$

where pos is the position and i is the dimension.

2.3 Attention Mechanism

Attention mechanism [11] is used to perform a mapping from a query to a series of key value pairs. There are three steps in the calculation of attention. The first step is to calculate the similarity between query and each key to get the weight. The common similarity functions are dot product, splicing, perceptron, etc.; the second step is to normalize these weights by using a softmax function; the last step is to sum the weight and the corresponding key value to get the final attention. The tokens in an article are first fed into the encoder to generate a sequence of encoder hidden states h_i, then the decoder receives the word embedding of previous step and derive the decoder state s_t at each step t. After that, the attentive context vector h_t^* can be obtained based on the attention distribution a^t as:

$$h_t^* = \sum_i a_i^t h_i, \tag{10}$$

$$a^t = \text{softmax}(e^t), \tag{11}$$

$$e^t = v^\top \tanh(W_h h_i + W_s s_t + b_{attn}), \tag{12}$$

where v, W_h, W_s, b_{attn} are learnable parameters. Finally, the predicted vocabulary distribution P_{vocab} is formulated as:

$$P_{vocab} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b'), \tag{13}$$

where V, V', b, b' are learnable parameters.

2.4 Copy and Pointing Mechanisms

The copy and pointing mechanisms proposed by PGN [93] is widely used in abstractive summarization, which is designed for alleviating the problem of inaccurate reproduced factual details via a pointer, dealing with out-of-vocabulary words and repetition via a generator and a coverage mechanism, respectively. In the pointer and generator, a generation probability p_{gen} based on the context vector h_t^*, decoder state s_t and decoder input x_t at step t is calculated by:

$$p_{gen} = \sigma(w_{h^*}^\top h_t^* + w_s^\top s_t + w_x^\top x_t + b_{ptr}), \tag{14}$$

which serves as a soft switch to choose between generating a word based on the vocabulary probability, or copying a word from the source document based on the attention distribution. The probability distribution over the extended vocabulary $P(w)$ can be formulated as:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i: w_i = w} a_i^t. \tag{15}$$

And in the coverage mechanism, a coverage vector c^t at step t is calculated by the sum of attention distributions over previous decoder timesteps as $c^t = \sum_{t'=0}^{t-1} a^{t'}$, which is then added to the attention mechanism and the primary loss function as:

$$e_i^t = v^\top \tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{attn}), \tag{16}$$

$$loss_t = -\text{log} P(w_t^*) + \lambda \sum_i \min(a_i^t, c_i^t), \tag{17}$$

where w_t^* is the target word at step t.

2.5 GAN

The generative adversarial network (GAN) [25] is a framework that uses an adversarial training process to estimate generative models. This framework can be regarded as a minimax two-player game containing a generative model and a discriminative model, and these two models are simultaneously trained. The generator (G) captures the data distribution and tries to produce fake samples, and the discriminator (D) attempts to determine whether the samples come from the model distribution or data distribution. In detail, G is trained to maximize the probability identified by D for the sample coming from the data rather than G, while D is trained to maximize the probability of assigning the correct label to training samples and samples generated by G. The training process continues until the counterfeits are indistiguishable from the genuine articles. The training objective with value function $V(D, G)$ can be formulated as:

$$\begin{aligned}\min_G \max_D V(D, G) =& \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] \\ &+ \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))].\end{aligned} \tag{18}$$

2.6 Memory Network

The end-to-end memory network [102] is based on the recurrent neural network. Before outputting a symbol, the recurrence reads from a possibly large external memory multiple times. In this framework, a discrete set of input sentences is written to the memory up to a fixed size, and a continuous representation for each memory sentence and the query (i.e., the question) is calculated and is then processed through multiple hops to output the answer. Specifically, for the single hop operation, the matching probability between each query and memory is first computed by the inner product followed by a softmax in the embedding space. The input set x and the query q are first embedded to obtain the memory vectors m and u, respectively, and the match probability p_i of the i-th input sentence is calculated as:

$$p_i = \text{Softmax}(u^\top m_i). \tag{19}$$

Then the output memory representation o is obtained by a weighted sum over another embedded memory sentences c based on the matching probability as:

$$o = \sum_i p_i c_i. \tag{20}$$

Finally, the final prediction $\hat{a}$ can be obtained through a weight matrix and a softmax over the sum of output memory vector o and input embedding u as:

$$\hat{a} = \text{Softmax}(W(o + u)). \tag{21}$$

To handle multiple hop operations, the memory layers are repeatedly stacked, and the input of each layer is the sum of the output memory vector and the input from its previous layer.

2.7 GNN

The graph neural network (GNN) [92] is used to process the data in graph form, which can capture the dependency information between nodes of a graph via message passing. Compared with CNN and RNN, GNN can propagate on each node respectively and is able to ignore the input orders of nodes, which is more computational efficient. The representation of each node in a graph is iteratively updated by aggregating information from its neighboring nodes and edges. So far, many propagation strategies have been proposed, such as convolution [38], RNN based gate mechanism [48], and attention mechanism [108]. Meanwhile, many works attempt to improve the training method, such as sampling-based training [28] and unsupervised training [37]. Take the graph convolution network [38] as an example, it follows the layer-wise propogation rule as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right), \tag{22}$$

where $\sigma(\cdot)$ refers to an activation fuction, $\tilde{A} = A + I_N$ is the adjacency matrix of the graph (including the identity matrix I_N), $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the degree of node i, $W^{(l)}$ is a trainable weight matrix in the l-th layer, and $H^{(l)}$ is the matrix of node feature vectors in the l-th layer.

Table 1. Natrual language generation models for text abbreviation.

Task	Model	Description
Text Summarization	MASS [99] BART [43] RCT [6] ProphetNet [80] En-semantic-model [15] Post-editing Factual Error Corrector [7] SpanFact [17]	PLM + Seq2Seq PLM + Seq2Seq PLM + RNN + CNN + Seq2Seq + Long-term Dependency PLM + Seq2Seq + Long-term Dependency RNN + Seq2Seq + Long-term Dependency PLM + Seq2Seq + Factual Consistency PLM + Seq2Seq + Factual Consistency
Question Generation	Key-phrase-based Question Generator [18] Dynamic Mathematical Question Generator [4] KB-based Factoid Question Generator [95] Teacher Forcing and RL Based Question Generator [130] Paragraph-level Question Generator [134] Answer-focused and Position-aware Question Generator [104] ASs2s [36] NQG-MP [113] Paraphrase Enhaced Question Generator [34] CGC-QG [53] PathQG [112]	Keyphrase + Template Constraint Handling Rules RNN + Seq2Seq RNN + Seq2Seq + RL RNN + Seq2Seq RNN + Seq2Seq + Answer-focused RNN + Seq2Seq + Answer-focused RNN + Seq2Seq + Multi-task RNN + Seq2Seq + Multi-task RNN + Seq2Seq + Multi-task + GNN RNN + Seq2Seq + Multi-task + KG
Distractor Generation	Educational Ontology Distractor Generator [101] Learning to Rank Based Distractor Generator [50] Hierarchical Dual-attention Distractor Generator [22] EDGE [82] HMD-Net [60] Code Compression Distractor Generator [100] CSG-DS [86] Named Entity Distractor Generator [73]	Ontology + Embedding Embedding + Ranking + GAN + RL RNN + Seq2Seq RNN + Seq2Seq + Answer Interaction PLM + RNN + Seq2Seq + Answer Interaction Abstract Syntax Tree LDA + KB + Ranking Tree + Clustering

2.8 Pre-trained Model

The pre-trained models can be divided into two categories: non-contextual and contextual. The non-contextual pre-trained models [63, 75] can learn high quality of word and phrase representations, and are widely used to initialize the embeddings and improve the model performance for generation tasks. However, since the non-contextual embeddings are static, it is unable to handle context-dependent words and out-of-vocabulary words. To overcome these problems, contextual pre-trained models are proposed [13, 77], which can dynamically change the word embeddings when the word appears in different sentences. The contextual models are mainly based on LSTM or Transformer architecture. For example, ELMo [77] uses deep bidirectional LSTM, BERT [13] uses the Transformer encoder. Meanwhile, more and more effective pre-training methods are proposed to improve the model's general representation ability. For example, BERT [13] presents a masked language model pre-training objective, MASS [99] designs a pre-training objective to guide the decoder predict the masked sentence fragments in the encoder side.

3 TEXT ABBREVIATION

3.1 Task

The goal of text abbreviation is to distill key information from long texts to short ones, which consists of three subtopics: text summarization, question generation, and distractor generation.

There are two kinds of methods for text abbreviation: extractive and abstractive methods. Since the abstractive approach is more flexible and can create more human-like sentences than the extractive approach, it has been paid more and more attentions in recent years and is our main focus in this paper. Text summarization is the process of generating entirely new phrases and sentences to capture the meaning of the source document. Question generation concentrates on automatically generating questions from a given sentence or paragraph. Distractor generation is the automatic generation of adequate distractors for a given question answer pair generated from a given article to form an adequate multiple-choice question.

3.2 Data

3.2.1 Text Summarization.

There are mainly four datasets in the field of text summarization as shown below.

CNN/DailyMail. The CNN/DailyMail dataset [29] is a large scale reading comprehension dataset. This dataset contains 93k and 220k articles collected from the CNN and Daily Mail websites, respectively, where each article has its matching abstractive summary.

NYT. The New York Times (NYT) dataset [74, 90] contains large amount of articles written and published by the New York Times between 1987 and 2007. In this dataset, most of the articles are manually summarized and tagged by a staff of library scientists, and there are over 650,000 article-summary pairs.

XSum. The extreme summarization (XSum) dataset [67] is an extreme summarization dataset containing BBC articles and corresponding single sentence summaries. In this dataset, 226,711 Wayback archived BBC articles are collected, which range from 2010 to 2017 and cover a wide variety of domains.

Gigaword. The English Gigaword dataset [26, 88] is a comprehensive collection of English newswire text data acquired by the Linguistic Data Consortium. This corpus contains four distinct international sources of English newswire, and has totally 4,111,240 documents.

3.2.2 Question Generation.

The two popular datasets for the task of question generation are shown below.

SQuAD. The Stanford Question Answering Dataset (SQuAD) [83] is a large reading comprehension dataset created by crowdworkers. The questions in this dataset are posed by crowdworkers based on a set of Wikipedia articles, and the answers are text segments from the corresponding passages. In total, SQuAD contains 107,785 question-answer pairs on 536 articles.

MS MARCO. The Microsoft Machine Reading Comprehension (MS MARCO) dataset [68] is a collection of anonymized search queries issued through Bing or Cortana for reading comprehension. The dataset contains both answerable and unanswerable questions. Each answerable question has a set of extracted passages from the retrieved response documents of Bing. There are totally 1,010,916 questions and 8,841,823 answering passages extracted from 3,563,535 web documents in this dataset.

3.2.3 Distractor Generation.

There are three datasets widely used for distractor generation as shown below.

SciQ. SciQ [116] is a crowdsourced multiple choice question answering dataset, which consists of 13.7k science exam questions. The domain of this dataset covers biology, chemistry, earth science, and physics.

MCQL. The MCQL dataset [50] is a collection of multiple choice questions at the Cambridge O level and college level, which is crawled from the Web. This dataset totally contains 7.1k questions covering biology, phisics, and chemistry.

RACE. RACE [41] is a reading comprehension dataset collected from the English exams in Chinese middle and high schools. This dataset contains 27,933 passages and 97,687 questions, covering all types of human articles.

3.3 Method

3.3.1 Text Summarization. Recently, the most common methods in this field are encoder-decoder based pre-trained language models. Song *et al.* [99] design a novel pre-training objective to jointly pre-train the encoder and decoder, where the decoder learns to predict the masked sentence fragments in the encoder side. Given an unpaired source sentence x from the source domain $\mathcal{X}$, the model with parameter θ predicts the sentence fragment $x^{u:v}$ from position u to v with the masked sequence $x^{\setminus u:v}$ as input, and the objective function is formualted as:

$$\begin{aligned} L(\theta;\mathcal{X}) &= \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log P(x^{u:v}|x^{\setminus u:v};\theta) \\ &= \frac{1}{|\mathcal{X}|}\sum_{xin\mathcal{X}} \log\prod_{t=u}^{v} P(x_t^{u:v}|x_{<t}^{u:v},x^{\setminus u:v};\theta) \end{aligned}, \tag{23}$$

Lewis *et al.* [43] present a denoising autoencoder for pre-training, which consists of a series of noising strategies to corrupt text and a training objective to reconstruct the original sentence.

However, the pre-trained language models' ability of capturing long-term dependencies and maintaining global coherence is poor. To solve this problem, Cai *et al.* [6] introduce an additional encoder with a bidirectional RNN and a convolution module to simultaneously model sequential context and capture local importance to the base Transformer model. The encoder first applies a bidirectional LSTM on the source text embedding E and derive the hidden states sequence H, then uses three convolution operations with kernel sizes of 1,3,5 to learn n-gram features D from H, and finally a gated linear unit is applied to select features, which can be formulated by:

$$R = \sigma(W_d D + b_d) \odot (W_h H + b_h), \tag{24}$$

$$H = BiLSTM(E). \tag{25}$$

Yan *et al.* [80] make several improvements to the traditional language models. Specifically, a new self-supervised objective called future n-gram prediction has been applied, and the n-stream self-attention mechanism is used in the decoder to be trained to predict the future n-gram of each time step. Given a source sequence x and its target sequence y, the future n-gram prediction objective can be formulated as:

$$\mathcal{L} = -\sum_{j=0}^{n-1} \alpha_j \cdot \left(\sum_{t=1}^{T-j} \log p_\theta(y_{t+j}|y_{<t},x)\right). \tag{26}$$

Ding *et al.* [15] propose an enhanced semantic model based on double encoders and a decoder with a Gain-Benefit gate structure, which is able to provide richer semantic information and reduce the influence of the length of generated texts on the decoding accuracy. The dual-encoder is used to capture both global and local context semantic information based on the bidirectional RNN, and the output vector of the Gain-Benefit gate module is calculated by:

$$a = \text{sigmoid}(\mathbf{W}_1\mathbf{C}_t + \mathbf{W}_2\mathbf{S}_{t-1} + y_{t-1}), \tag{27}$$

$$\mathbf{P}_t = (1-a)\odot\mathbf{M} + a\odot\mathbf{C}_{t-1}. \tag{28}$$

where $\mathbf{C}_t$ is the contextual semantic representation with both global and local semantic information at current time step, $\mathbf{S}_{t-1}$ is the decoder hidden state of last time step, y_{t-1} is the word generated at last time step, and $\mathbf{M}$ is the global semantic vector of the original text.

Except for the poor capability of capturing long-term dependency in traditional pre-trained language models, the problem of factual inconsistency between the generated content and source text is also severe and has not been tackled by previous works. To reduce this phenomenon, Meng *et al.* [7] propose an end-to-end neural corrector model with a post-editing correction strategy, which is pre-trained on artificial corrupted reference summaries. Dong *et al.* [17] introduce a factual correction framework containing a QA-span factual correction model and an auto-regressive one. The QA-span correction model masks and replaces one entity at a time during the iteration process. Specifically, given the source text x and a masked query $q = (y'_1, ..., [MASK], ..., y'_m)$, the correction model needs to predict the answer span via $p(i = start)$ and $p(i = end)$, which are calculated based on the hidden states of the top layer h_i as:

$$p(i = start) = a_i^{start} = \frac{\exp(q_i^s)}{\sum_{j=0}^{H-1} \exp(q_j^s)}, \tag{29}$$

$$q_i^s = \mathrm{ReLU}(w_s^\top h_i + b_s), \tag{30}$$

where H is the number of hidden states in the encoder, and $p(i = end)$ is calculated in the similar way. The auto-regressive correction model masks all entities at the same time without iteration. Specifically, given the source text x and a masked query $q = (y'_1, ..., [MASK]_1, ..., [MASK]_T, ..., y'_m)$ from a summary with T entities, the correction model runs T steps to predict the answer span of each mask based on the corresponding masked token representation and its previously predicted entity representations. The entity representation $\mathbf{s}_t^{ent}$ at time step t is predicted based on the argmax and mean pooling operations, as calculated by:

$$\mathbf{s}_t^{ent} = \text{Mean-Pool}(\mathbf{h}_{p_{start}}, \mathbf{h}_{p_{end}}), \tag{31}$$

$$p_{start} = \arg\max(a_1^{start}, ..., a_M^{start}), \tag{32}$$

$$p_{end} = \arg\max(a_1^{end}, ..., a_M^{end}). \tag{33}$$

3.3.2 Question Generation. Early methods to solve the task of question generation are mainly based on hand-crafted rules and always contain multiple procedures. Wijanarko *et al.* [18] propose a method that generates questions based on key-phrase which embeds Bloom's taxonomic in selecting contexts for constructing questions. The key-phrases can be selected via two methods. The first method is based on probability of two-word sequence as formulated below:

$$score(w_i, w_j) = \frac{count(w_i w_j) - \delta}{count(w_i) \times count(w_j)}, \tag{34}$$

where $count(w_i, w_j)$ is the number of a sequence of word w_i followed by w_j, $count(w_i)$ is the number of word w_i in the input document, and δ is a constant to limit the number of phrases formed by less frequent words. The second method is based on a Naive Bayes model, and the probability of a unique phrase is measured by:

$$Pr[key|T, D] = \frac{Pr[T|key] \times Pr[D|key] \times Pr[key]}{Pr[T, D]}, \tag{35}$$

where $Pr[T|key]$ is the probability that a key-phrase has a $TF{\times}IDF$ score T, $Pr[D|key]$ is the probability that it has a distance D, and $Pr[T, D]$ is the normalization factor. Bhatia *et al.* [4] propose a dynamic question answering generator to generate questions and answers for the mathematical topic-quadratic equations. The randomization technique, first order logic and automated deduction are used for the study.

In the past few years, driven by advances in deep learning, end-to-end neural models based on Seq2Seq framework have attained more and more attentions and have shown better performances.

Serban *et al.* [95] study the factoid questions generation with RNN to transduce facts into neural language questions, and provide an enormous question-answer pair corpus. Yuan *et al.* [130] use a Seq2Seq model with teacher forcing to improve the training and adopts policy gradient in reinforcement learning to optimize the generated results. During supervised learning, in addition to minimize the negative log-likelihood with teacher forcing, two additional signals are introduced to prevent the model from generating answer words ($\mathcal{L}_s$) and encourage the output variety ($\mathcal{L}_e$) as formulated below:

$$\mathcal{L}_s = \lambda_s \sum_t \sum_{\bar{a} \in \bar{\mathcal{A}}} p_\theta(y_t = \bar{a} | y_{<t}, D, A), \tag{36}$$

$$\mathcal{L}_e = \lambda_e \sum_t \mathbf{p}_t^T \text{log} \mathbf{p}_t, \tag{37}$$

where $\bar{\mathcal{A}}$ refers to the set of words appearing in the answer but not in the ground-truth question, $\mathbf{p}_t$ refers to the full pointer-softmax probability of the t-th word, which enables the model to interpolate between copying from the source document and generating from shortlist. And during reinforcement learning, the total reward R_{PPL+QA} is a combination of question answering reward R_{QA} (measured by F1 score) and question fluency reward R_{PPL} (measured by perplexity), as formulated below:

$$R_{PPL+QA} = \lambda_{QA} R_{QA}(\hat{Y}) + \lambda_{PPL} R_{PPL}(\hat{Y}), \tag{38}$$

$$R_{PPL}(\hat{Y}) = -2^{-\frac{1}{T}\sum_{t=1}^{T} \log_2 PLM(\hat{y}_t | \hat{y}_{<t})}, \tag{39}$$

$$R_{QA}(\hat{Y}) = \text{F1}(\hat{A}, A), \tag{40}$$

where $\hat{A} = MPCM(\hat{Y})$ refers to the answer of the generated question by the Multi-Perspective Context Matching (MPCM) model, *PLM* is a language model. Zhao *et al.* [134] study the paragraph-level neural question generation by proposing a maxout pointer and gated self-attention networks, which mainly deals with the problem that long text (mostly paragraphs) does not perform well in the Seq2Seq model. In detail, the gated self-attention network contains two steps: 1) the encoded passage-answer representation $\mathbf{u}_t$ is conducted matching against itself to derive the self matching representation $\mathbf{s}_t$ at time step t:

$$\mathbf{s}_t = \mathbf{U} \cdot \mathbf{a}_t^s, \tag{41}$$

$$\mathbf{a}_t^s = softmax(\mathbf{U}^\top \mathbf{W}^s \mathbf{u}_t). \tag{42}$$

And 2) the self matching representation $\mathbf{s}_t$ is combined with the original passage-answer representation $\mathbf{u}_t$, which is then fed into a feature fusion gate to obtain the final encoded passage-answer representation $\hat{\mathbf{u}}_t$ at time step t:

$$\hat{\mathbf{u}} = \mathbf{g}_t \odot \mathbf{f}_t + (1 - \mathbf{g}_t) \odot \mathbf{u}_t, \tag{43}$$

$$\mathbf{f}_t = tanh(\mathbf{W}^f [\mathbf{u}_t, \mathbf{s}_t]), \tag{44}$$

$$\mathbf{g}_t = sigmoid(\mathbf{W}^g [\mathbf{u}_t, \mathbf{s}_t]). \tag{45}$$

However, the rich information lying in the answer has not been fully explored, which leads to generating low-quality questions (e.g., having mismatched interrogative words with the answer type, copying context words far and irrelevant to the answer, including words from the target answer, and so on). To solve the problem, Sun *et al.* [104] propose an answer-focused and position-aware model. It incorporates the answer embedding to explicitly generate a question word matching the answer type, and designs a position-aware attention mechanism by modeling the relative distance with the answer, which guides the model to copy the context words that are more close and relevant to the answer. Kim *et al.* [36] propose to separate the target answer from the original passage, in order to avoid the generated question copying words from the answer. Specifically, it replaces the answer with a mask token, and introduces a keyword-net to extract key information from the answer. Given the encoded answer representation h^a and the context vector c_t of current decoding step, the keyword feature of each layer of the keyword-net o_t^l can be formulated as:

$$o_t^l = \sum_j p_{tj}^l h_j^a, \tag{46}$$

$$p_{tj}^l = Softmax((o_t^{l-1})^\top h_j^a), \tag{47}$$

where o_t^0 is initialized by c_t. And the decoding hidden state s_t of current timestep is calculated by:

$$s_t = LSTM(y_{t-1}, s_{t-1}, c_t, o_t^L), \tag{48}$$

where y_{t-1} is the output token of previous timestep, L is the layer number of the keyword-net.

Moreover, many works recently attempt to conduct multi-task learning with external related tasks to further enhance the performance of question generation. Wang *et al.* [113] introduce a message passing mechanism to simultaneously learn the tasks of phrase extraction and question generation, which helps the model be aware of question-worthy phrases that are worthwhile to be asked about. Jia *et al.* [34] conduct multi-task learning with paraphrase generation and question generation, which can diversify the question patterns of the question generation module. During training, the weights of the encoder are shared by all tasks while those of the first layer of decoder are shared with a soft sharing strategy, which is formulated by:

$$\mathcal{L}_{sf} = \sum_{\mathrm{d} \in \mathcal{D}} \|\theta_d - \phi_d\|_2, \tag{49}$$

where $\mathcal{D}$ is a set of shared decoder parameters, θ, ϕ refer to the parameters of the question generation task and paraphrase generation task, respectively. And a min-loss function is employed among the golden reference question and several expanded question paraphrases, which is represented by:

$$\mathcal{L}_{qg} = \min_{\mathbf{q} \in Q}(-\frac{1}{T_{qg}} \sum_{t=1}^{T_{qg}} logP(y_t^{qg} = \mathbf{q}_t)). \tag{50}$$

Despite the outstanding performance achieved by previous methods, the rich structure information hidden in the passage is ignored, which can be used as an auxiliary knowledge of the unstructured input text to improve the performance. Liu *et al.* [53] adopt a graph convolutional network (GCN) to identify the clue words in the input passage that should be copied into the target question. Specifically, the GCN is constructed on the syntactic dependency tree representation of each passage, and a Gumbel-Softmax layer is applied to the final representation of GCN to sample the binary clue indicator for each word. A sample $\mathbf{y} = (y_1, ..., y_k)$ drawn from the Gumbel-Softmax distribution is formulated as:

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^k \exp((\log(\pi_j) + g_j)/\tau)}, \tag{51}$$

where τ is the temperature parameter, π_i is the unnormalized log probability of class i, and g_i is the Gumbel noise formulated by:

$$g_i = -\log(-\log(u_i)), \tag{52}$$

$$u_i \sim \text{Uniform}(0, 1). \tag{53}$$

Wang *et al.* [112] construct a knowledge graph for each input sentence as the auxiliary structured knowledge and aims to generate a question based on a query path from the knowledge graph. The query representation learning is formulated as a sequence labeling problem for identifying the involved facts to form a query, which is used to generate more relevant and informative questions.

3.3.3 Distractor Generation. The researches mainly focus on generating multi-choice question distractors for ontologies or articles. Traditional methods primarily use hand-crafted rules or ranking method for distractor generation. Stasaski *et al.* [101] introduce a novel method with several ontology- and embedding-based approaches. The graph structure of the ontology is used to create complex problems linking different concepts. Liang *et al.* [50] introduce a ranking method with a feature-based model and a neural net (NN) based model. The NN-based model consists of a generator G and a discriminator D, where G generates distractors d based on a conditional probability $P(d|q,a)$ given question stems q and answers a, and D predicts whether a distractor sample comes from the real training data or G. The objective for D is to maximize the log-likelihood as:

$$\begin{aligned} \max_{\phi}\mathbb{E}_{d\sim P_{true}(d|q,a)}&[\log(\sigma(f_{\phi}(d|q,a)))] \\ + \mathbb{E}_{d\sim P_{\theta}(d|q,a)}&[\log(1-\sigma(f_{\phi}(d|q,a)))], \end{aligned} \tag{54}$$

where $f_{\phi}(d,q,a)$ is an arbitrary scoring function parameterized by ϕ. And each distractor d_i sampled by G is based on another scoring function $f_{\theta}(d,q,a)$ as formulated as:

$$p_{\theta}(d_i|q,a) = \frac{\exp(\tau \cdot f_{\theta}(d_i,q,a))}{\sum_j \exp(\tau \cdot f_{\theta}(d_j,q,a))}, \tag{55}$$

where τ is a temperature hyper-parameter. Then a cascaded learning framework is proposed to make the ranking more effective, which divides the ranking process into two stages to reduce the candidates.

Recently, Seq2Seq model is widely adopted due to its overwhelming performance. Gao *et al.* [22] express the task as a sequence-to-sequence learning problem based on a hierarchical encoder-decoder network. In this model, static and dynamic attention mechanisms are adopted on the top of the hierarchical encoding structure, and a question-based initializer is used as the start point to generate distractors in the decoder. The question q, the answer a and the word vectors in the i-th sentence $(\mathbf{w}_{i,1},...,\mathbf{w}_{i,m})$ are first encoded via three separate bidirectional LSTM networks into $(\mathbf{q}_1,...,\mathbf{q}_l)$, $(\mathbf{a}_1,...,\mathbf{a}_k)$, and $(\mathbf{h}^{e}_{i,1},...,\mathbf{h}_{i,m})$. Then another bidirectional LSTM is applied on the encoded word representations to derive the contextualized sentence representation $(\mathbf{u}_1,...,\mathbf{u}_n)$, and an average pooling layer is applied to derive their entire representations $\mathbf{q}$, $\mathbf{a}$, and $\mathbf{s}_i$. After that, the static attention distribution γ_i can be derived through a matching layer and a normalization layer as:

$$\gamma_i = \text{softmax}(o_i/\tau), \tag{56}$$

$$\tau = \text{sigmoid}(\mathbf{w_q}^{\top}\mathbf{q} + b_q), \tag{57}$$

$$o_i = \lambda_q \mathbf{s}_i^{\top}\mathbf{W}_m\mathbf{q} - \lambda_a \mathbf{s}_i^{\top}\mathbf{W}_m\mathbf{a} + \mathbf{b}_m. \tag{58}$$

In the decoder size, the decoder generates the hidden state $\mathbf{h}^d_t$ at the t-th time step through an LSTM network. Then the sentence-level and word-level dynamic attention β_i and $\alpha_{i,j}$ are formulated as:

$$\beta_i = \mathbf{u}_i^{\top}\mathbf{W}_{d_1}\mathbf{h}^d_t, \quad \alpha_{i,j} = \mathbf{h}^{e}_{i,j}{}^{\top}\mathbf{W}_{d_2}\mathbf{h}^d_t. \tag{59}$$

Finally, the static and dynamic attentions are combined into $\tilde{\alpha}_{i,j}$ to reweight the article token representations and predict the probability distribution P_V over vocabulary V:

$$P_V = \text{softmax}(\mathbf{W}_V \tilde{\mathbf{h}}_t^d + \mathbf{b}_V), \tag{60}$$

$$\tilde{\mathbf{h}}_t^d = \tanh(\mathbf{W}_{\tilde{\mathbf{h}}}[\mathbf{h}_t^d; \mathbf{c}_t]), \tag{61}$$

$$\mathbf{c}_t = \sum_{i,j} \tilde{\alpha}_{i,j} \mathbf{h}_{i,j}^e, \tag{62}$$

$$\tilde{\alpha}_{i,j} = \frac{\alpha_{i,j}\beta_i\gamma_i}{\sum_{i,j}\alpha_{i,j}\beta_i\gamma_i}. \tag{63}$$

However, the answer interaction is not considered by previous works and the incorrectness of the generated distractors cannot be guaranteed. To address this problem, Qiu *et al.* [82] propose a framework consisting of reforming modules and an attention-based distractor generator. The reforming modules use the semantic distances to constrain the effect of words that are strongly related to the correct answer, and the distractor generator leverages the information of the reformed question and passage to generate the initial state and context vector respectively. In detail, three contextual encoders are first applied to encode the passage, question and its answer into **P**, **Q** and **A**, then an attention mechanism and a fusion kernel are leveraged to enrich the question and answer representations into $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{A}}$, where $\tilde{\mathbf{Q}}$ is formulated by:

$$\tilde{\mathbf{Q}} = \text{Fuse}(\mathbf{Q}, \bar{\mathbf{Q}}) = \tanh([\mathbf{Q}; \bar{\mathbf{Q}}; \mathbf{Q} - \bar{\mathbf{Q}}; \mathbf{Q} \circ \bar{\mathbf{Q}}]\mathbf{W}_f + \mathbf{b}_f), \tag{64}$$

$$\bar{\mathbf{Q}} = \text{Attn}(\mathbf{Q}, \mathbf{P})\mathbf{P} = \text{softmax}(\frac{\mathbf{Q}\mathbf{P}^T}{\sqrt{d}})\mathbf{P}. \tag{65}$$

In the reforming question module, the reformed question $\dot{\mathbf{Q}}_i$ is calculated through a self-attend layer and a gate layer as:

$$\dot{\mathbf{Q}} = \text{Gate}(\tilde{\mathbf{Q}}_i, \tilde{\mathbf{v}}^a)\tilde{\mathbf{Q}}_i = (\tilde{\mathbf{Q}}_i \mathbf{W}_g^q \tilde{\mathbf{v}}^{a^\top} + b_g^q)\tilde{\mathbf{Q}}_i, \tag{66}$$

$$\tilde{\mathbf{v}}_a = \text{SelfAlign}(\tilde{\mathbf{A}}) = \text{softmax}(\tilde{\mathbf{A}}\mathbf{W}_a)^\top \tilde{\mathbf{A}}. \tag{67}$$

And in the reforming passage module, the reformed passage $\tilde{\mathbf{P}}$ is calculated by:

$$\tilde{\mathbf{P}} = \text{Fuse}(\dot{\mathbf{P}}, \bar{\mathbf{P}}),\ \bar{\mathbf{P}} = \text{Attn}(\dot{\mathbf{P}}, \dot{\mathbf{Q}})\dot{\mathbf{Q}},\ \dot{\mathbf{P}}_i = \text{Gate}(\mathbf{P}_i, \hat{\mathbf{v}}^a)\mathbf{P}_i, \tag{68}$$

$$\hat{\mathbf{v}}^a = \text{SelfAlign}(\hat{\mathbf{A}}),\ \hat{\mathbf{A}} = \text{Fuse}(\tilde{\mathbf{A}}, \bar{\mathbf{A}}),\ \bar{\mathbf{A}} = \text{Attn}(\tilde{\mathbf{A}}, \tilde{\mathbf{Q}})\tilde{\mathbf{Q}}. \tag{69}$$

Maurya *et al.* [60] use a single encoder to encode the input triplet and three decoders to generate three distractors. The encoder employs SoftSel operation and a gated mechanism to capture the semantic relations among the elements of the input triplet.

In addition, there are also many other scenarios for distractor generation. Srinivas *et al.* [100] develop a semi-automatic tool to help teachers quickly create multiple choice questions on code understanding. The tool first captures each code structure in the form of an abstract syntac tree, and then train a code model that maps functions to vectors. Ren *et al.* [86] create a model based on a context-sensitive candidate set generator and a distractor selector for cloze-style multiple choice questions. The model first uses the correct answer as the key, combines the LDA model to mine the topic of the context, finds similar words in the semantic database, and then selects a specified number of misleading items according to a ranking model. Specifically, the probability distribution over all entites subsumed by the concepts in C is calculated based on the posterior

Table 2. Natrual language generation models for text expansion.

Task	Model	Description
Short Text Expansion	Associated-query-based Query Expander [5] ExpaNet [106] FC-LSTM [96] Fiction Sentence Expander [89]	Ranking Retrieval + Memory Network LDA + RNN + Ranking RNN + Seq2Seq
Topic-to-Essay Generation	MTA-LSTM [19] SRENN [114] UD-GAN [129] KB-based Topic-to-essay Generator [123] SCTKG [81]	RNN + Seq2Seq + Global Coherence RNN + Seq2Seq + Retrieval + Global Coherence RNN + GAN + RL CNN + RNN + Seq2Seq + GAN + RL + KB CNN + RNN + Seq2Seq + GAN + RL + KG

probability $p(c|a,q)$ as:

$$p_i = p(d_i|a,q) \propto \sum_{c \in C} p(d_i|c)p(c|a,q), \tag{70}$$

$$p(c|a,q) \propto p(c|a) \sum_{k=1}^{K} \pi_{a,q}^{(k)} \gamma_c^{(k)}, \tag{71}$$

where c is the concept, $\pi_{a,q}$ is the topic distribution of complete sentence formed by the stem and key, γ_c is the topic distribution of concept c, $p(c|a)$ is the prior probability of a belonging to c, K is the total number of topics, and $p(d|c)$ is the typicality. Patra *et al.* [73] develop a system to generate named entity distractors. The system performs two types of similarity computation namely statistical and semantic. To speed up the distractor selection procedure, a hierarchical clustering method is proposed to represent the entities, where the entity similarities are embedded in a tree structure. The distractors are selected from the nearby entities of the correct answer of the question in the tree. Specifically, the statistical distance between the numeric attributes is calculated by:

$$Sim(P,Q) = 1 - \frac{1}{L} \sum_{i=1,\ldots,L} \frac{(P_i \sim Q_i)}{\max(P_i, Q_i)}, \tag{72}$$

where P and Q represent two vectors corresponding to the target entities, L is the total number of numeric attributes. The hierarchical distance between two entities (x,x') is normalized by:

$$Sim(x,x') = \frac{d_1(x,x')}{\sqrt{d_1(x,x) \odot d_1(x',x')}}, \tag{73}$$

where $d_1(x,x')$ is the highest tree level connecting x and x'. And the semantic similarity score between the key x and a candidate distractor x' is formulated as:

$$Sim(x,x') = \frac{|(triplet_i \in x)|\&(triplet_i \in x')|_i}{|triplet_j \in x|_j}, \tag{74}$$

where the normalization factor is the size of the triplet set corresponding to the key.

4 TEXT EXPANSION

4.1 Task

The main purpose of text expansion is to inflate the short texts to longer ones that contains more abundant information, which can be divided into two aspects: short text expansion and topic-to-essay generation. Short text expansion aims to expand a short text into a richer representation based on a set of long documents. Topic-to-essay generation aims at generating human-like diverse, and topic-consistent paragraph-level text with a set of given topics.

4.2 Data

4.2.1 Short Text Expansion.

There are mainly four datasets for short text expansion as listed below.

Wikipedia. [106] constructs this dataset through a snapshot of English Wikipedia. The titles of Wikipedia articles are regarded as short texts, and the abstract of all Wikipedia articles are leveraged to construct the related long documents. This dataset contains 30,000 short texts and 4,747,988 long documents in total.

DBLP. [106] uses the DBLP bibliography database to construct this dataset. The titles of computer science literature represent short texts, and the abstracts of all papers are collected to construct the corresponding long documents. Statistically, this dataset consists of 81,479 short texts and 480,558 long documents.

Programmableweb. [96] establishes a real-world dataset for service recommendation and description expansion, which is crawled from programmableweb.com in 2013 and 2016. The entire dataset contains 16012 APIs, 7816 Mashups and 16449 links between them.

Fiction Corpus. [89] creates an English fiction corpus, which is obtained by applying sentence compression techniques on a modern fiction corpus scraped from online resources. Each story sentence has a corresponding compression. In total, the dataset contains around 17,000,000 sentences.

4.2.2 Topic-to-Essay Generation.

There are primarily five datasets for topic-to-essay generation as listed below.

ESSAY. The ESSAY dataset [19] is a large collection of topic compressions crawled from the Internet. The topic words are extracted by TextRank. This datast totally contains 305,000 paragraph-level essays.

ZhiHu. [19] constructs this dataset by crawling from a Chinese question-and-anwering website called ZhiHu, which consists of 55,000 articles. The topic words of each article are specified by users in the community.

Moview Reviews. The Stanford Sentiment Treebank (SST) dataset [98] contains 11,855 single sentences of movie reviews. This dataset has two sentiment classes for each review, and has a set of fully labeled parse trees.

Beer Reviews. [61] creates this dataset by crawling from the beer review website BeerAdvocate. In total, this dataset has 1,586,259 ratings on 66,051 items that are scored by 33,387 users.

Customer Reviews. [31] constructs a customer review dataset consisting of five electronics products, which is collected from Amazon.com and C|net.com. There are totally 1,886 items in this dataset.

4.3 Method

4.3.1 Short Text Expansion.

Early works primarily use statistical similarity for short text expansion. Billerbeck *et al.* [5] propose a method about the query expansion using associated queries for web search engines. The main method is to associate a query that closely matches the document in a given log containing a large number of queries. Then, the query associated with the query document is a reasonable description of the document and can be used to expand the query text.

Benefit from the development of deep learning, many end-to-end frameworks based on neural networks have been developed. Tang *et al.* [106] propose an end-to-end solution based on deep memory network for short text extension. In this work, the original short text q is firstly used as a query to search for a set of potentially relevant long documents C_q, which will be used as the material for text expansion, from an external large collection C. In the following process, the short text q is represented as the average vector of words in it, i.e. $\overrightarrow{q}$, and each document is also represented by the average vector of words belong to it, i.e. $\overrightarrow{d}$. To further identify the relevant documents in C_q, both soft attention and hard attention mechanism are utilized, the information read from the document set can be written as:

$$\overrightarrow{o} = \sum_{i=1}^{K} p_i \overrightarrow{d_i}, \tag{75}$$

$$p_i = \text{softmax}(\frac{\overrightarrow{q}^T \overrightarrow{d_i} + g_i}{\tau}), \tag{76}$$

where g_i follows the Gumbel(0,1) distribution, and τ is the temperature hyperparameter. Then the two sources of information, $\overrightarrow{q}$ and $\overrightarrow{o}$ are integrated by GRU as follows:

$$\overrightarrow{z} = \sigma(\mathbf{W}^{(z)}\overrightarrow{q} + \mathbf{U}^{(z)}\overrightarrow{o}), \tag{77}$$

$$\overrightarrow{r} = \sigma(\mathbf{W}^{(r)}\overrightarrow{q} + \mathbf{U}^{(r)}\overrightarrow{o}), \tag{78}$$

$$\overrightarrow{o}' = \tanh(\mathbf{W}\overrightarrow{q} + \overrightarrow{r} \circ \mathbf{U}\overrightarrow{o}), \tag{79}$$

$$\overrightarrow{q}' = (1 - \overrightarrow{z}) \circ \overrightarrow{q} + \overrightarrow{z} \circ \overrightarrow{o}', \tag{80}$$

where $\circ$ means element-wise multiplication, both the Sigmoid function $\sigma(x)$ and $tanh(x)$ are operated on element-wise. The output $\overrightarrow{q}'$ is the expanded representation of the input short text q. This process can be repeated several times for the same q to continuously extend its representation to mimic the human behavior when querying a piece of short text.

Shi *et al.* [96] present a text expansion method for service recommendation system. The description of services is first expanded at sentence level by a probabilistic topic model, in this process, the similarity between target sentence u and another sentence v from existed corpus can be caculated as:

$$Similarity(u,v) = \mu \cdot D_{JS}(u,v) + (1-\mu) \cdot D_{JS}(S_u, S_v), \tag{81}$$

where S_u and S_v are descriptions contain u and v, respectively. μ is a parameter used to balance weights of sentence and description on the final similarity measurement. D_{JS} is the function of JS divergence which used to measure the similarity between two items; for more details, readers can refer to [96]. So for each target sentence, a collection of similar sentences can be found and ranked in descent order to select top N most similar ones for description extension.

Safovish *et al.* [89] design a neural sentence expander which is trained on a corpus of fiction sentence compressions for the task of sentence expansion and enhancement. In this method, a Seq2Seq model is used to predict sentences from the original input. To tackle the problem of copying input words during generation process, the modified negative log-likelihood loss function is used to increase the significance of learning new words. Specifically, the modified cross-entropy of generating is caculated as:

$$-\sum_t (1 + \lambda \mathrm{I}_{T-S}(w_t)) \log p(w_t | w_1, ..., w_{t-1}), \tag{82}$$

in which I denotes the indicator function for a set, T the ground truth, S the source tokens, λ is a parameter which controls the learning importance of the words in target sentence while not in original input. With this modification, the model no longer degenerates to copying its input and can be trained as desired. In addition, to increase the novelty of generated sentence, Safovish *et al.* [89] also develop a controlled sampling method for the decoding process so that the model can generate diverse output.

4.3.2 Topic-to-essay Generation. With the popularity of deep learning, the RNN-based Seq2Seq framework has been widely used in this field. Many works have been proposed to improve the traditional Seq2Seq framework.

To improve the global coherence of the generated essays, Feng *et al.* [19] propose an LSTM model with attention mechanism for essay generation. The main idea of this work is to maintain a topic coverage vector, each dimension of which represents the degree to which a topic word needs to be expressed in future generation, to adjust the attention policy, so that the model can consider more about unexpressed topic words. Specifically, the topic coverage vector is updated by parameter ϕ_j, and it can be regarded as the discourse-level weight of $topic_j$, the word embedding of topic word i. Topic coverage vector C_t is initialized as a k dimensional vector, i.e. C_0, k is the number of input topic words, and each value of C_0 is 1.0. At time step t in generation process, each element $c_{t,j}$ is updated as follows:

$$c_{t,j} = c_{t-1,j} - \frac{\alpha_{t,j}}{\phi_j}, \tag{83}$$

in which $\alpha_{t,j} = \exp(g_{tj})/\sum_{i=1}^{K} \exp(g_{ti})$ is the attention weight of topic word i at time step t, and g_{tj} is the attention score on $topic_j$ at time step t, which is caculated as:

$$g_{tj} = c_{t-1,j} v_a^T \tanh(W_a h_{t-1} + U_a topic_j), \tag{84}$$

where h_{t-1} is the hidden representation of the LSTM at time step $t-1$, and v_a, W_a, U_a are all parameters to be optimized. Therefore, the probability of the next word y_t can be defined as:

$$\mathrm{P}(y_t|y_{t-1}, T_t, C_t) = \mathrm{softmax}(\mathrm{linear}(h_t)), \tag{85}$$

where the topic representation T_t is formulated as $T_t = \sum_{i=1}^{K} \alpha_{tj} topic_j$.

Wang *et al.* [114] propose an enhanced neural network based on self-attention and retrieval mechanisms, the encoder and decoder are constructed with self-attention to model longer dependence. And to alleviate the duplication problem, a retrieval process is adopted to collect topic-related sentences as an aid for essay generation. Specifically, input topic words are divided into m groups, then the material $M = \{S_1, ..., S_m\}$ can be collected based on cosine distance between the topic group and sentences in the corpus which is created by dividing the training set, where S_i is a sentence that corresponds to the i-th topic group. For the different sequence relations between topic words and material sentences, two encoders with the same structure but different parameters are used to get the hidden representation $H^{Topic} = \{h_1^T, ..., h_k^T\}$ and $H^{Material} = \{h_1^M, ..., h_{ml}^M\}$, based on which the hidden states of generated of essay words $H = \{h_1, ..., h_n\}$ are obtained by essay decoder. Finally the output probabilities of each essay word can be computed as:

$$p(y_t|y_{t-1}, T, M) = \mathrm{softmax}(\mathrm{linear}(h_t)). \tag{86}$$

Meanwhile, Generative Adversarial Net (GAN) has shown promising results for topic-to-essay generation, which is effectively applied in [129] which proposes a GAN model with two-level discriminators. The first discriminator guides the generator to learn the paragraph-level information and sentence syntactic structure with multiple LSTMs, and the second one processes higher level information such as topic and sentiment for essay generation. Then the reward is calculated based on results of two discriminators, and the generator G_θ tries to maximize expected reward from the initial state till the end state via the formulation:

$$J(\theta) = \sum_{t=1}^{T} E(R_t|S_{t-1}, \theta) = \sum_{t=1}^{T} G_\theta(y_t|Y)[\lambda(D_\phi(Y)) + (1-\lambda)D_\gamma(Y)], \tag{87}$$

where λ is a manually set weight, Y is a complete sequence, R_t is the reward for a whole sequence, D_ϕ and D_γ are the first and second discriminator, respectively.

However, the generated essays of previous works are still lack of novelty and topic-consistency, based on which, some works leverage the external knowledge to solve this problem. Yang *et al.* [123] introduce a model with the aid of commonsense knowledge in ConceptNet, in detail, each topic is used as a query to retrive k neighboring concepts with pre-trained embeddings stored as commonsense knowledge in a memory matrix M_0, in the decoding phase, the generator G_θ refers to the memory matrix for text generation, the hidden state of the decoder at time-step t is:

$$s_t = \mathrm{LSTM}(s_{t-1}, [e(y_{t-1}); c_t; m_t]), \tag{88}$$

where ; means the concatenation of vectors, y_{t-1} is the word generated at time-step $t-1$. c_t is the context vector that is computed by integrating the hidden representations of the input topic sequence, and m_t is the memory vector extracted from $\mathbf{M}_t$ based on the attention mechanism. As the generation progresses, the topic information that needs to be expressed keeps changing, which requires the memory matrix to be dynamically updated, so for each memory entry $\mathbf{M}_t^i$ in $\mathbf{M}_t$, a candidate update memory $\widetilde{\mathbf{M}}_t^i$ is computed as:

$$\widetilde{\mathbf{M}}_t^i = \tanh(\mathbf{U}_1\mathbf{M}_t^i + \mathbf{V}_1 e(y_t)), \tag{89}$$

where $\mathbf{U}_1$ and $\mathbf{V}_1$ are trainable parameters. To determine how much the i-th memory entry should be updated, the adaptive gate mechanism is adopted:

$$g_t^i = \mathrm{sigmoid}(\mathbf{U}_2\mathbf{M}_t^i + \mathbf{V}_2 e(y_t)), \tag{90}$$

where $\mathbf{U}_1$ and $\mathbf{V}_1$ are trainable parameters. $\mathbf{M}_t^i$ is updated by:

$$\mathbf{M}_{t+1}^i = (\mathbf{1} - g_t^i) \odot \mathbf{M}_t^i + g_t^i \odot \widetilde{\mathbf{M}}_t^i, \tag{91}$$

in which $\mathbf{1}$ refers to the vector with all elements 1 and $\odot$ denotes point-wise multiplication. Following their work, Qiao *et al.* [81] propose a topic-to-essay generator based on the conditional variational auto-encoder framework to control the sentiment, and introduces a topic graph attention mechanism to sufficiently use the structured semantic information which is ignored in [123], so the quality of essay generation is further improved.

5 TEXT REWRITING AND REASONING

5.1 Task

The target of text rewriting and reasoning is to make a reversion of the text or apply reasoning methods to generate responses, which mainly contains two subtopics: text style transfer and dialogue generation.

Text style transfer is the method to transform the attribute style of the sentence while preserving its attribute-independent content. Dialogue generation aims to automatically generate approximate answers to a series of given questions in a dialogue system.

5.2 Data

5.2.1 Text Style Transfer. The following four datasets are widely applied for the task of text style transfer.

Table 3. Natrual language generation models for text rewriting and reasoning.

Task	Model	Description
Text Style Transfer	Spearean Modern Language Translator [33]	RNN + Seq2Seq
	ARAE [133]	RNN + GAN + WAE + Seq2Seq
	FM-GAN [9]	RNN + GAN + Seq2Seq
	Unpaired Sentiment-to-sentiment Translator [120]	RNN + Seq2Seq + RL + Unsupervised
	Non-offencive Language Translator [69]	CNN + RNN + Seq2Seq + Unsupervised
	BST [79]	CNN + RNN + Seq2Seq + Unsupervised
	DualRL [58]	RNN + Seq2Seq + RL + Unsupervised
	Exploration-evaluation-based Text Style Translator [20]	RNN + Seq2Seq + Evaluation Metrics Design
	Evaluating Style Transfer for Text [64]	Embedding + CNN + RNN + Evaluation Metrics Design
Dialogue Generation	MARM [135]	RNN + Seq2Seq
	GAN-AEL [122]	CNN + RNN + Seq2Seq + GAN
	AEM [59]	RNN + Seq2Seq + Context-response Interaction
	AGMN [46]	RNN + KG + Retrieval + Context-response Interaction
	Post-KS [49]	RNN + Seq2Seq + KB
	KADG [35]	Transformer + Seq2Seq + KB
	Bert2trans [115]	PLM + Seq2Seq
	PLATO [3]	PLM + Seq2Seq

Yelp Review. The Yelp Review dataset is provided by the Yelp Dataset Challenge[1], which contains a large amount of business review texts. This dataset contains 1.43M, 10K, and 5K pairs for training, validation, and testing, respectively.

Amazon Food Review. [61] creates this dataset by crawling reviews from the Fine Foods category of Amazon. This dataset contains 367K, 10K, and 5K pairs for training, validation, and testing, respectively.

EMNLP2017 WMT News. [27] picks the News section data from the EMNLP2017 WMT[2] Dataset, which is a large long-text corpus consists of 646,459 words and 397,726 sentences. After being preprocessed, this dataset contains 278,686 and 10,000 sentences for training and testing, respectively.

GYAFC. The Grammarly's Yahoo Answers Formality Corpus (GYAFC) [84] is a large corpus for formality stylistic transfer. The informal sentences are collected from Yahoo Answers[3] with domains of Entertainment & Music and Family & Relationships, and the corresponding formal sentences are created using Amazon Mechanical Turk. This dataset totally contains 106,000 sentence pairs.

5.2.2 *Dialogue Generation.*

There are four popular datasets for dialogue generation as shown below.

DailyDialog. The DailyDialog dataset [47] is a high-quality multi-turn dialog dataset containing daily conversations. The dialogues in the dataset is formally written by human with reasonable speaker turns, and often concentrate on a certain topic. Statistically, this dataset contains 13,118 multi-turn dialogues, nearly 8 average speaker turns per dialogue, and about 15 average tokens per utterance.

UDC. The Ubuntu Dialogue Corpus (UDC) [56] is created based on the two-person conversations about Ubuntu-related problems in the Ubuntu chat logs[4] from 2004 to 2015. This dataset contains about 1 million multi-turn dialogues, over 7 million utterances, and 100 million words.

[1] https://www.yelp.com/dataset/

[2] http://statmt.org/wmt17/translation-task.html

[3] https://answers.yahoo.com/answer

[4] http://irclogs.ubuntu.com/

Persona-chat. The Persona-chat dataset [131] is an engaging and personal chit-chat dialogue dataset collected by Amazon Mechanical Turk. Each of the paired crowdworkers condition their dialogue on a given provided profile. This dataset contains 164,356 utterances in total.

Wizard-of-Wikipedia. The Wizard-of-Wikipedia dataset [14] is a large crowd-sourced collection for open-domain dialogue. Each of the paired speakers conduct open-ended chit-chat, and one of the speakers need to link the knowledge to sentences from existing Wikipedia articles connected to the topic. This dataset contains 22,311 dialogues with 201,999 turns.

5.3 Method

5.3.1 Text Style Transfer. In this field, the RNN-based Seq2Seq framework and deep latent variable model are broadly adopted. Jhamtani *et al.* [33] design a copy-enriched sequence-to-sequence model to transform text from modern English to Shakespearean English. The model first uses a fixed pre-trained embedding vector to represent each token, and uses a bidirectional LSTM to encode sentences. In this work,$\overrightarrow{LSTM_{enc}}$ and $\overleftarrow{LSTM_{enc}}$represent the for- ward and reverse encoder.$h_t^{\overrightarrow{enc}}$represent hidden state of encoder model at step t.The following equations describe the model:

$$h_t^{\overrightarrow{enc}} = \overrightarrow{LSTM_{enc}}(h_{t-1}^{enc}, E_{enc}(x_t)), \tag{92}$$

$$h_t^{\overleftarrow{enc}} = \overleftarrow{LSTM_{enc}}(h_{t+1}^{enc}, E_{enc}(x_t)), \tag{93}$$

$$h_t^{enc} = h_t^{\overrightarrow{enc}} + h_t^{\overleftarrow{enc}}. \tag{94}$$

In this work, only the forward and backward encoder states are added, and the standard connection is not used because it does not add additional parameters.Then a mixture model of RNN and pointer network are employed to transfer the text style. The pointer module provides location based attention, and output probability distribution due to pointer network module can be expressed as follows:

$$p_t^{PTR}(w) = \sum_{x_j = w} (\beta_j). \tag{95}$$

Zhao *et al.* [133] propose an adversarially regularized autoencoder framework to generalize the adversarial autoencoder, which combines a discrete autoencoder with a regularized latent representation of GAN, and can be further formalized by the Wasserstein autoencoder.The model is trained with coordinate descent across: (1)the encoder and decoder to minimize reconstruction, (2) the critic function to approximate the W term, (3) the encoder adversarially to the critic to minimize W:

$$1) \min_{\phi,\psi} L_{rec}(\phi,\psi) = E_{X\sim P_*}[-\log p_\psi(x|enc_\psi(x))], \tag{96}$$

$$2) \max_{w\in W} L_{cri}(w) = E_{X\sim P_*}[f_w(enc_\psi(x))] - E_{\widetilde{z}\sim P_z}[f_w(\widetilde{z})], \tag{97}$$

$$3) \min_{\phi} L_{enc}(\phi) = E_{X\sim P_*}[f_w(enc_\psi(x))] - E_{\widetilde{z}\sim P_z}[f_w(\widetilde{z})]. \tag{98}$$

Here enc_ψ is a deterministic encoder function. P_z is the the prior distribution, and $f_w(\widetilde{z})$ is the critic/discriminator.

Chen *et al.* [9] utilize optimal transport to improve the ability of traditional GAN in processing descrete texts with the objective of feature-mover's distance. In this work, the feature-mover's distance(FMD) between two sets of sentence features is then defined as:

$$D_{FMD}(P_f, P_{f'}) = \min_{T\geq 0} \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij} \cdot c(f_i, f_{j'}) = \min_{T\geq 0} < T, C >, \tag{99}$$

where $\sum_{j=1}^{n} T_{ij} = \frac{1}{m}$ and $\sum_{i=1}^{m} T_{ij} = \frac{1}{n}$ are the constraints, and $<, >$ represents the Frobenius dot-product. In this work, the transport cost is defined as the cosine distance: $c(f_i, f_{j'}) = 1 - \frac{f_i^T f_j^{'}}{\|f_i\|_2 \|f_i^{'}\|_2}$ and C is the cost matrix.

However, the problem of lacking supervised parallel data has not been well studied by the above works. To tackle this problem, many unsupervised methods have been proposed. Xu *et al.* [120] propose a cycled reinforcement learning approach through the cooperation between the neutralization and emotionalization modules. The neutralization module extracts the non-emotional part of the sentence with a single LSTM and a self-attention based sentiment classifier, and the emotionalization module generates emotional words and add them to the semantic content with a bi-decoder based encoder-decoder framework.

Santos *et al.* [69] propose an unsupervised style transfer model to convert offensive language into non-offensive one. A single collaborative classifier is used to train the encoder-decoder network, and an attention mechanism with a cycle consistency loss is adopted to preserve the content. Prabhumoye *et al.* [79] realize the style transfer of gender, political slant and sentiment through an unsupervised back-translation method. The transferring process is divided into two stages. The first stage learns the latent representation with back-translation of the input sentence through a language translation model, and the second stage adopts an adversarial generation technology to enable the output to match the desired style.The latent representation with back-translation of the input sentence can be described as:

$$z = Encoder(X_f; \theta_E), \tag{100}$$

where, x_f is the sentence x in language f. θ_E represent the parameters of the encoder of language $f \longrightarrow$ language e translation system

Luo *et al.* [58] reformulate the traditional unsupervised style transferring task as a one-step mapping problem, and propose a dual reinforcement learning framework to train the source-to-target and target-to-source mapping models. In this work, two reward methods that can evaluate style accuracy and content preservation separately are proposed. $R_s = P(s_y|y^{'}; \psi)$ formulate the style classifier reward where ψ is the parameter of the classifier and is fixed during the training process, and $R_c = P(x|y^{'}; \phi)$ represents the reward for preserving content. To encourage the model to improve both the content preservation and the style accuracy, the final reward is the harmonic mean of the above two rewards:

$$R = (1 + \beta^2) \frac{R_c \cdot R_s}{(\beta^2 \cdot R_c) + R_s}, \tag{101}$$

where β is a harmonic weight aiming to control the trade-off between the two rewards.

Moreover, another challenge in this field is that there does not exist reliable evaluation metrics, which is neglected by previous works. To alleviate this issue, Fu *et al.* [20] propose two aspects of evaluation metrics to measure the transfer strength and content preservation of style transfer. The transfer strength aims to evaluate whether the style is transferred through an LSTM-sigmoid classifier. The style is defined in (30). This classifier is based on keras examples2. Transfer strength accuracy is defined as $\frac{N_{right}}{N_{total}}$, $N_{total}$is the number of test data, and $N_{right}$is the number of correct case which is transferred to target style.

$$l_{style} = \begin{cases} paper(positive) & output \leq 0.5 \\ news(negative) & output \geq 0.5 \end{cases}. \tag{102}$$

The content preservation is used to evaluate the similarity between source and target texts and is calculated by the embedding cosine distance.Content preservation rate is defined as cosine distance (31) between source sentence embedding v_s and target sentence embedding v_t.

$$score = \frac{v_s^T v_t}{||v_s|| \cdot ||v_t||}. \tag{103}$$

Mir *et al.* [64] specify three aspects of evaluation metrics including style transfer intensity, content preservation, and naturalness. The style transfer intensity is measured by Earth Mover's Distance, the content preservation is calculated by METEOR and embedding-based metrics, and the naturalness is obtained by an adversarial evaluation method.

5.3.2 Dialogue Generation. The RNN-based or GAN-based Seq2Seq model is widely leveraged to handle this task. Zhou *et al.* [135] propose a mechanism-aware neural machine based on a probabilistic RNN-based Seq2Seq framework. The model first uses latent embeddings to represent the corresponding mechanisms, then an encoder-diverter-decoder framework is leveraged to generate mechanism-aware context. In this study, there are M latent mechanisms $M_i{}_{i=1}^{M}$ for response generation. Then, $p(y|x)$ can be expanded as follows:

$$p(y|x) = \sum_{i=1}^{M} p(y, m_i|x) = \sum_{i=1}^{M} p(m_i|x)p(y|m_i, x), \tag{104}$$

where $p(m_i|x)$ represents the probability of the mechanism m_i conditioned on x. This probability actually measures the degree that m_i can generate the response for x. The bigger of this value is, the more degree that the mechanism m_i can be used to generate the responses for x. Additionally, $p(y|m_i, x)$ measures the probability that the response y is generated by the mechanism m_i for x. With the modeling of $p(m_i|x)$ and $p(y|m_i, x)$ the objective of likelihood maximization, namely

$$\sum_{(x,y)\in D^c} \log p(y|x) = \sum_{(x,y)\in D^c} \log \sum_{i=1}^{M} p(m_i|x)p(y|m_i, x), \tag{105}$$

is used to learn the mechanism embeddings $M_i{}_{i=1}^{M}$ and other model parameters.

Xu *et al.* [122] introduce a GAN framework comprising a generator, a discriminator, and an approximate embedding layer to generate informative responses. The generator uses a Seq2Seq model with GRU to generate responses, and the discriminator uses a convolutional neural network to judge the difference between human responses and machine responses. In the approximate embedding layer, the overall word embedding approximation is computed as:

$$\hat{e}_{w_i} = \sum_{j=1}^{V} e_j \cdot Softmax(W_p(h_i + Z_i) + b_p)j, \tag{106}$$

where w_p and b_p are the weight and bias parameters of the word projection layer, respectively, and h_i is the hidden representation of word w_i, from the decoding procedure of the generator G.

However, previous works ignore the semantic and utterance relationships between the context and response, and thus the obtained responses are not satisfying. To solve this problem, Luo *et al.* [59] propose an auto-encoder matching model with a mapping module. In this model, two auto-encoders are leveraged to learn the semantic representations, and the mapping module is used to learn the utterance-level dependency between the context and response. In this mapping module, for simplicity, there is only a simple feedforward network for implementation. The mapping module M_γ transforms the source semantic representation h to a new representation t. To be specific, we implement a multi-layer perceptron (MLP) $g(\cdot)$ for M_γ and train it by minimizing the L2-norm loss $J_3(\gamma)$ of the transformed representation $t = g(h)$ and the semantic representation of target response s:

$$J_3(\gamma) = \frac{1}{2}\|t - s\|_2^2. \tag{107}$$

Li *et al.* [46] design a dual encoder model with an attention mechanism and a graph attention network. The attention mechanism is responsible of capturing the relationship between context and response, and the graph attention network is used to integrate the knowledge connections of domain words. The concept representation in the domain knowledge is constructed by a series of triples, $G(x) = \{T1, T2, ..., Tn\}$ where T_i has the same concept node u but different neighbor concept v and the graph representation of the concept $g(x)$ can be calculated by graph attention mechanism as:

$$g(x) = \sum_{i=1}^{n} \alpha T_i [u_i^e; v_i^e], \tag{108}$$

$$\alpha T_i = \frac{exp(\beta_{T_i})}{\sum_{j=1}^{n} exp(\beta_{T_i})}, \tag{109}$$

$$\beta_{T_i} = Relu([(u_i^e)^T W v_i^e]), \tag{110}$$

where $(u_i, r_i, v_i) = R_i \in G(x)$is the i-th triple in the dataset.

Some researchers try to improve the response quality by incorporating external knowledge bases, and propose many strategies to select appropriate knowledge. Lian *et al.* [49] present a knowledge selection mechanism by separating the posterior distribution from the prior distribution. The distance between the posterior and prior distributions are minimized by the KL divergence during training, and during inference, the knowledges are selected and incorporated into the response based on the prior distribution.The Kullback-Leibler divergence loss (KLDivLoss), to mea- sure the proximity between the prior distribution and the posterior distribution, which is defined as follows:

$$L_{KL}(\theta) = \sum_{i=1}^{N} p(k = k_i|x, y) \log \frac{p(k = k_i|x, y)}{p(k = k_i|x)}, \tag{111}$$

where θ denotes the model parameters.

Jiang *et al.* [35] propose a knowledge augmented response generation model to improve the knowledge selection and incorporation. The model consists of a divergent knowledge selector and a knowledge aware decoder, where the selector conducts a one-hop subject reasoning over facts to reduce the subject gap in the knowledge selection, and the decoder is used to efficiently incorporate the selected fact.

Recently, pre-trained language models (e.g., BERT) have shown significant improvements over traditional RNN-based methods in many NLP tasks and are also applied to this task. Wang *et al.* [115] propose an encoder-decoder framework containing a BERT encoder and a transformer decoder. The encoder is used to learn semantic representations for both unstructured text and conversational history, and the decoder is leveraged to generate the dialogue response.In the encoder of this work,the input embedding is the sum of its token embedding, knowledge indicating embedding and position embedding:

$$I(x_i) = E(x_i) + T(x_i) + P(x_i), \tag{112}$$

where $E(x_i)$, $T(x_i)$, $P(x_i)$ are word embedding, knowledge indication embedding and position embedding, respectively. The input embeddings are then fed into BERT model to get the knowledge and dialogue history encoding representations.

Bao *et al.* [3] design a pre-training framework with discrete latent variables. The pre-training tasks include repsonse generation and latent act recognition, which are jointly pre-trained through a unified network with shared parameters.

Table 4. From Image to Text Generation

Task	Model	Description
Image Caption	Show and Tell [110]	CNN + LSTM
	BUTD [1]	Faster-RCNN + LSTM
	Knowing When to Look [57]	CNN + LSTM
	Exploring Visual Relationship for Image Captioning [126]	Faster-RCNN + LSTM + GCN
	Self-critical Sequence Training [87]	CNN + LSTM + RL
	Auto-Encoding Scene Graphs [125]	CNN + LSTM + GCN + RL
Video Caption	LRCN [16]	CNN + LSTM
	S2VT [109]	CNN + LSTM + Knowledge
	Dense Caption Events [118]	CNN + LSTM + Daps
	Masked Transformer [136]	CNN + TCN + Transformer
	Hierarchical reinforcement learning [111]	CNN + LSTM + RL
	Adversarial Inference [72]	CNN + LSTM + Discriminator
	VideoBert Pretrain [103]	CNN + Bert
	ActBert Pretrain [137]	CNN + Bert + Multi-task Learning
	ClipBert [42]	CNN + Bert + Clip Sampling
Visual Storytelling	Informative Visual Storytelling [45]	CNN + GRU
	Knowledgeable Storyteller [124]	CNN + GRU + Graph
	Composite reward [30]	CNN + RNN + MLE

6 FROM IMAGE TO TEXT GENERATION

6.1 Task

The image-based text generation aims at explaining or summarizing the visual concept of the given image, which is mainly consisted of three parts: image caption, video caption, and visual storytelling. The purpose of image captioning is to generate summaries from an image. Based on image caption, video caption aims to generate the summary of a series of images. Visual storytelling not only identifies the correlation between objects in a single picture but also gives the logical relationship between consecutive sequential images. It should be noted that although VQA is a relatively similar topic to image caption, there exists the main distinction that current VQA systems are mainly designed to choose answers from a given candidate answer set, which is not related to the natural language generation.

6.2 Data

6.2.1 Image Caption. The literature review of the image caption datasets is shown below.

Flickr30k. [78] contains 31,783 images collected from Flickr. Most of these images depict humans performing various activities. Each image is paired with 5 crowd-sourced captions.

COCO. [52] is the largest image captioning dataset, containing 82,783, 40,504 and 40,775 images for training, validation and test respectively. This dataset is more challenging, since most images contain multiple objects in the context of complex scenes. Each image has 5 human annotated captions.

Visual Genome. [40] is composed of dense annotations of objects, attributes, and relationships within each image to learn these models. Specifically, this dataset contains over 108K images where each image has an average of 35 objects, 26 attributes, and 21 pairwise relationships between objects.

6.2.2 Video Caption. There are mainly three popular datasets for video caption as shown below.

MSR-VTT. [119] is the most widely used video caption dataset, which contains 7,180 videos of 20 categories. This is created by collecting 41.2 hours of 10K web video clips from a commercial video search engine.

Charades. [97] is collected with a focus on common household activities using the Hollywood in Homes approach. This dataset contains 9,848 videos with 66,500 annotations describing 157 actions.

ActivityNet. [39] is a large dataset that connects videos to a series of temporally annotated sentences. Each sentence describes what occurs in an unique segment of a video. Specifically, AcitvityNet contains 20k videos with 100k sentence-level description.

6.2.3 Visual Storytelling. The following two datasets are widely used for visual storytelling.

VIST. [32] is the most widely used dataset for visual storytelling. It contains 10,032 visual albums with 50,136 stories. Each story contains five narrative sentences, corresponding to five grounded images respectively.

VideoStory. [24] contains 20k videos posted publicly on a social media platform amounting to 396 hours of video with 123k sentences.

6.3 Method

6.3.1 Image Captioning. Image captioning aims to generate the description for an image automatically. Image captioning [110] has achieved significant improvements based on neural encoder-decoder framework. The Show-Tell model [110] employs convolutional neural networks to encode image into fixed-length vector, and recurrent neural networks as decoder to sequentially generate words. To capture fine-grained visual details, attentive image captioning models [57] are proposed to dynamically ground words with relevant image parts in generation. Based on the consideration that region aware feature better fits the human visual system, Anderson *et al.* propose a recognized baseline called BUTD [1]. The BUTD caption model is composed of two LSTM layers. The first LSTM layer is designed to capture the top-down visual attention model, while the second LSTM layer is regard as a language model. Given the mean-pooled image feature $\overline{\boldsymbol{v}}$, a word embedding matrix W_e, and a one-hot encoding Π_t of the input word at time t we could obtain the output $\boldsymbol{h}_t^1$ as:

$$\boldsymbol{h}_t^1 = \text{LSTM}\left(\left[\boldsymbol{h}_{t-1}^2, \overline{\boldsymbol{v}}, W_e\Pi_t\right], \boldsymbol{h}_{t-1}^1\right), \tag{113}$$

and the normalized attention weight $a_{i,t}$ can be represented by the following formulations:

$$a_{i,t} = \boldsymbol{w}_a^T \tanh\left(W_{va}\boldsymbol{v}_i + W_{ha}\boldsymbol{h}_t^1\right), \tag{114}$$

$$\boldsymbol{\alpha}_t = \text{softmax}\left(\boldsymbol{a}_t\right), \tag{115}$$

The attended image feature used as input to the language LSTM is calculated as a convex combination of all input features as $\hat{v}_t = \sum_{i=1}^{K} \alpha_{i,t} v_i$. The input to the language model LSTM consists of the attended image feature, concatenated with the output of the attention LSTM, given by $\boldsymbol{h}_t^2 = \left[\hat{v}_t, \boldsymbol{h}_t^1\right]$. Using the notation $y_{1:T}$ to refer to a sequence of words, at each time step t the conditional distribution over possible output words is given by:

$$p\left(y_t \mid y_{1:t-1}\right) = \text{softmax}\left(W_p\boldsymbol{h}_t^2 + \boldsymbol{b}_p\right), \tag{116}$$

The distribution over complete output sequences is calculated as the product of conditional distributions:

$$p\left(y_{1:T}\right) = \prod_{t=1}^{T} p\left(y_t \mid y_{1:t-1}\right). \tag{117}$$

To reduce exposure bias and metric mismatching in sequential training, notable efforts are made to optimise non-differentiable metrics using reinforcement learning [54, 87, 121]. To further boost accuracy, detected semantic concepts [21, 117, 127] are adopted in captioning framework. A more structured representation over concepts calling scene graph is further explored [125, 126] in image captioning which can take advantage of detected objects and their relationships. Instead of using a fully detected scene graph to improve captioning accuracy, Chen *et al.* [10] propose to employ Abstract Scene Graph as control signal to generate intention-aware and diverse image captions.

6.3.2 Video Caption. The aim of video caption is to describe or summarize a video in natural language. It is a non-trivial task for computers since it is difficult to select the useful visual features from a video clip and describe what's happening in a way that obeys the common sense of humanity.

The currently prevailing architecture for video caption is composed of a CNN-like visual encoder and a RNN-like linguistic decoder. Donahue *et al.* [16] design the Long-term Recurrent Convectional Networks (LRCNs) that is both temporally and spatially deep. After that, Venugopalan *et al.* [109] introduce the S2VT, a Seq2Seq approach for video to text with the knowledge from text corpora. To capture the dependencies between the events in a video, Krishna *et al.* [39] propose a captioning module that uses contextual information from past and future events. However, Krishna *et al.* [39] fail to take advantage of language to benefit event proposal with the co-training diagram. Thus, Zhou *et al.* [136] propose an end-to-end framework for video caption that is able to produce proposal and description simultaneously. To describe a video with multiple fine-grained actions, Wang *et al.* [111] propose a hierarchical reinforcement learning framework that a high-level agent learns to design sub-goals and a low-level worker recognizes the primitive actions to fulfill the sub-goal. By introducing the discriminator that considers visual relevance to the video, language diversity & fluency, and coherence across sentences, Park *et al.* [72] generate more accuracy video descriptions. To resolve the dilemma that encoders of vision-language tasks are not trained end to end, Lei *et al.* [42] propose ClipBert, a framework that applies the sparse sampling to use a few sampled clips to achieves better performance.

Recent years, self-supervised learning has become increasingly important with its power to leverage the abundance of unlabeled data. Sun *et al.* [103] propose VideoBert to learn bidirectional joint distributions over sequences of visual and linguistic tokens without any explicit supervision. Zhu *et al.* [137] propose ActNet, which models global and local visual cues for fine-grained visual and linguistic relation learning.

6.3.3 Visual Storytelling. Visual Storytelling not only needs to identify the correlation between objects in a single picture, but also Need to identify and learn the logical relationship between consecutive sequential images. In practice, Visual Storytelling is prone to problems such as single narrative words, rigid sentences, incoherent context logic, and lack of emotion in story descriptions. aims at generating a coherent and reasonable story with a series of images [32]. To deal with the issue that Visual Storytelling usually focus on generating general description rather than the details of meaningful visual contents, Li *et al.* [45] propose to mine the cross-modal rules to assistant the concept inference. Yang *et al.* [124] present a commonsense-driven generative model to introduce crucial commonsense from the external knowledge base for visual storytelling. Due to the limitation of maximum likelihood estimation on training, the majority of existing models encourage high resemblance to texts in the training database, which makes the description overly rigid and lack in diverse expressions. Therefore, Mo *et al.* [65] cast the task as a reinforcement learning problem and propose an Adversarial All-in-one Learning (AAL) framework to learn a reward model, which simultaneously incorporates the information of all images in the photo stream and all texts in the paragraph, and optimize a generative model with the estimated reward. To make the

Visual Storytelling model topic adaptively, Li *et al.* [45] introduce a gradient-based meta-learning algorithm. Previous storytelling approaches mostly focused on optimizing traditional metrics such as BLEU, ROUGE and CIDEr. In this paper, Hu *et al.* [30] revisit the issue from a different perspective, by delving into what defines a natural and thematically coherent story.

7 NLG EVALUATION METRICS

In this section, we mainly introduce several automatic evaluation metrics for NLG, which can be divided into two categories: untrained evaluation metrics, and machine-learned evaluation metrics [8].

7.1 Untrained Evaluation Metrics

This category of metric is most widely used in the NLG community since it is easy to be implemented and does not involve additional training cost, which compares machine-generated texts to human-generated ones simply based on content overlap, string distance or lexical diversity. We mainly introduce five metrics of such category, including BLEU, ROUGE, METEOR, Distinct, and Self-BLEU.

The Bilingual Evaluation Understudy (BLEU) metric [71] is used to calculate the co-occurrence frequency of two sentences based on the weighted average of matched n-gram phrases. BLEU was originally used to evaluate machine translation, and has been used for more and more NLG tasks, such as question generation [134], topic-to-essay generation [123], text style transfer [58], and dialogue generation [3].

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric [51] is used to measure the similarity between the generated and reference texts based on the recall score. This metric is commonly used in the field of text summarization, including four types: ROUGE-n measures the n-gram co-occurrence statistics; ROUGE-l measures the longest common subsequence; ROUGE-w measures the weighted longest common subsequence; ROUGE-s measures the skip-bigram co-occurrence statistics. ROUGE has also been widely applied to other NLG tasks such as question generation [134], distractor generation [82], and dialogue generation [3].

The Metric for Evaluation of Translation with Explicit Ordering (METEOR) metric [2] is an improvement over BLEU to address several weaknesses including four aspects: lack of recall, use of higher order n-grams, lack of explicit word-matching between translation and reference, and use of geometric averaging of n-grams, which is calculated by the harmonic mean of the unigram precision and recall. In addition to machine translation, METEOR has also been widely used in text summarization [80], question generation [134], and dialogue generation [3].

The Distinct metric [44] is used to measure the diversity of response sequences for dialogue generation. It calculates the number of distinct unigrams and bigrams in generated responses to reflect the diversity degree. To avoid preference for long sequences, the value is scaled by the total number of generated tokens.

The Self-BLEU metric [138] is also a metric to measure the diversity. Different from BLEU that only evaluates the similarity between two sentences, Self-BLEU is used to measure the resemblance degree between one sentence (hypothesis) and the rest sentences (reference) in a generated collection. It first calculates the BLEU score of every generated sentence against other sentences, then the average BLEU score is defined as the Self-BLEU score of the document, where a lower Self-BLEU score implies higher diversity.

7.2 Machine-learned Evaluation Metrics

This category of metric is based on machine-learned models to simulate human judges, which evaluates the similarity between machine-generated texts or between machine-generated texts and human-generated ones. We mainly introduce three metrics of such category, containing ADEM, BLEURT, and BERTScore.

The Automatic Dialogue Evaluation Model (ADEM) metric [55] is used to automatically evaluate the quality of dialogue responses, where the evaluation model is trained in a semi-supervised manner with a hierarchical recurrent neural network (RNN) to predict the response scores. Specifically, given the dialogue context **c**, model response **r̂**, and reference response **r** encoded by a hierarchical RNN, the predicted score can be calculated by:

$$score = (\mathbf{c}^\top M\hat{\mathbf{r}} + \mathbf{r}^\top N\hat{\mathbf{r}} - \alpha)/\beta, \tag{118}$$

where M, N are learnable matrices initialized by the identity, and α, β are scalar constants to initialize the predicted scores in range [1,5].

The Bilingual Evaluation Understudy with Representations from Transformers (BLEURT) metric [94] is based on BERT [13] with a novel pre-training scheme. Before fine-tuning BERT on rating data to predict human rating scores, a pre-training method is applied, where BERT is pre-trained on a large number of synthetic sentence pairs on several lexical- and semantic-level supervision signals in a multi-task manner. This pre-training process is important and can improve the robustness to quality drifts of generation systems.

The BERTScore metric [132] uses pre-trained contextual embeddings from BERT to measure the similarity between two sentences. Given the contextual embeddings of a reference sentence x and a candidate sentence $\hat{x}$, namely $\mathbf{x}, \hat{\mathbf{x}}$, the recall, precision, and F1 scores are calculated by:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \tag{119}$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \tag{120}$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}, \tag{121}$$

where the recall is calculated by matching each token in x to a token in $\hat{x}$, the precision is obtained by matching each token in $\hat{x}$ to a token x, and greedy matching is adopted to match the most similar tokens.

7.3 Human Evaluation Metrics

For generation, human evaluation focuses on the explanation of two key matters: diversity and creativity, i.e., the capacity of varying their texts in form and emphasis to fit an enormous range of speaking situations, and the potential to express any object or relation as a natural language text. In further detail, human evaluation is implemented to evaluate on three aspects: Grammar (whether a generated sentence is fluent without grammatical error), Faithful (whether the output is faithful to input), and Coherent (whether a sentence is logically coherent and the order of expression is in line with human writing habits). This needs to organize the capabilities of the people who work on generation in field of computational linguistics and artificial intelligence.

8 PROBLEMS AND CHALLENGES

In this section, we primarily point out four problems and challenges that deserve to be tackled and investigated further, including the evaluation method, external knowledge engagement, controllable generation, and multimodal scenarios.

Evaluation Method. Evaluation method is still an important and open research area for the field of NLG. As pointed by [12], traditional untrained evaluation metrics do not always correlate well with human judgements, while recent machine-learned metrics need a large amount of human annotations and not always have good transferability. Hence, there still exists a significant amount of challenges and improvement room in this area.

External Knowledge Engagement. Considering the limited information lying in the original texts and the difficulty of generating satisfying sentences [128], it is crucial to incorporate external knowledge to enhance the performance. Therefore, how to obtain useful and correlative knowledge and how to effectively incorporate the knowledge still deserve to be investigated.

Controllable Generation. Another challenging problem is how to generate controllable natural language as we would like it to be. Although a great body of work has been done in this area to study how to perform various kinds of controlled text generation, there is still a lack of uniform paradigms and standards about it. More importantly, how to measure the controllability of the generated text remains an open question, for different controlled contents.

Multimodal Scenarios. Recently, research on various applications in multimodal scenarios have gradually attracted more and more attention from NLP researchers. How to apply natural language generation methods in multimodal scenarios has been a worthy problem and promising direction. It is reasonable to believe that the utilization of rich multimodal information into natural language generation tasks will surely further advance the progress and development in this direction.

9 CONCLUSIONS

Over the past few years, natural language generation tasks and methods have become important and indispensable in natural language processing. This progress owes to advances in various deep learning-based methods. This article describes deep learning research on natural language generation with a historical perspective, emphasizing the special character of the problems to be solved. It begins by contrasting generation with language understanding, establishing basic concepts about the tasks, datasets, and the deep learning methods through it. A section of evaluation metrics from the output of generation systems follows, showing what kinds of performance are possible and where the difficulties are. Finally, some open problems are suggested to indicate the major challenges and future research directions of natural language generation.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[2] 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.*

[3] Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*

[4] Rahul Bhatia, Vishakha Gautam, and Yash Garg. 2019. Dynamic Question Answer Generator: An Enhanced Approach to Question Generation. *International Journal of Trend in Scientific Research and Development* (2019).

[5] Bodo Billerbeck, Falk Scholer, Hugh E. Williams, and Justin Zobel. 2003. Query Expansion Using Associated Queries. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management.*

[6] Tian Cai, Mengjun Shen, Huailiang Peng, Lei Jiang, and Qiong Dai. 2019. Improving Transformer with Sequential Context Representations for Abstractive Text Summarization. In *Natural Language Processing and Chinese Computing.*

[7] Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual Error Correction for Abstractive Summarization Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.*

[8] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2021. Evaluation of Text Generation: A Survey. In *arXiv preprint arXiv:2006.14799.*

[9] Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial Text Generation via Feature-Mover's Distance. In *Advances in Neural Information Processing Systems.*

[10] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say As You Wish: Fine-Grained Control of Image Caption Generation With Abstract Scene Graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[11] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-Based Models for Speech Recognition. In *Advances in Neural Information Processing Systems.*

[12] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. (2021).

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[14] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations.*

[15] Jianli Ding, Yang Li, Huiyu Ni, and Zhengquan Yang. 2020. Generative Text Summary Based on Enhanced Semantic Attention and Gain-Benefit Gate. *IEEE Access* (2020).

[16] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, and Kate Saenko. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[17] Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-Fact Correction in Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.*

[18] Bambang Dw, Yaya Heryadi, Hapnes Toba, and Widodo Budiharto. 2021. Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy. *Education and Information Technologies* (2021).

[19] Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-Essay Generation with Neural Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.*

[20] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style Transfer in Text: Exploration and Evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[21] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic Compositional Networks for Visual Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[22] Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R. Lyu. 2019. Generating Distractors for Reading Comprehension Questions from Real Examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[23] Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *Journal of Artificial Intelligence Research* (2018).

[24] Spandana Gella, Mike Lewis, and Marcus Rohrbach. 2018. A Dataset for Telling the Stories of Social Media Videos. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

[25] 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.*

[26] David Graff and Christopher Cieri. 2003. English Gigaword. LDC2003T05. Web Download. Philadelphia: Linguistic Data Consortium.

[27] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence.*

[28] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems.*

[29] 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.*

[30] Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2020. What Makes A Good Story? Designing Composite Rewards for Visual Storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence* (2020).

[31] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[32] Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual Storytelling. In *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies.

[33] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing Modern Language Using Copy-Enriched Sequence to Sequence Models. In *Proceedings of the Workshop on Stylistic Variation.*

[34] Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. How to Ask Good Questions? Try to Leverage Paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*

[35] Bin Jiang, Jingxu Yang, Chao Yang, Wanyue Zhou, Liang Pang, and Xiaokang Zhou. 2020. Knowledge Augmented Dialogue Generation with Divergent Facts Selection. *Knowledge-Based Systems* (2020).

[36] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving Neural Question Generation Using Answer Separation. *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).

[37] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. In *arXiv preprint arXiv:1611.07308.*

[38] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations.*

[39] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *IEEE International Conference on Computer Vision.*

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* (2017).

[41] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.*

[42] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less Is More: ClipBERT for Video-and-Language Learning via Sparse Sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[43] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*

[44] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[45] Jiacheng Li, Haizhou Shi, Siliang Tang, Fei Wu, and Yueting Zhuang. 2019. Informative Visual Storytelling with Cross-Modal Rules. In *Proceedings of the 27th ACM International Conference on Multimedia.*

[46] Xu Li and Jinghua Zhu. 2020. Attention and Graph Matching Network for Retrieval-Based Dialogue System with Domain Knowledge. In *2020 International Joint Conference on Neural Networks.*

[47] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing.*

[48] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *International Conference on Learning Representations.*

[49] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.*

[50] Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor Generation for Multiple Choice Questions Using Learning to Rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications.*

[51] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out.*

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision.*

[53] Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to Generate Questions by Learning What Not to Generate. In *The World Wide Web Conference.*

[54] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved Image Captioning via Policy Gradient optimization of SPIDEr. In *IEEE International Conference on Computer Vision.*

[55] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

[56] 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. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue.*

[57] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[58] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.*

[59] Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

[60] Kaushal Kumar Maurya and Maunendra Sankar Desarkar. 2020. Learning to Distract: A Hierarchical Multi-Decoder Network for Automated Generation of Long Distractors for Multiple-Choice Questions for Reading Comprehension. In *the 29th ACM International Conference on Information & Knowledge Management.*

[61] Julian John McAuley and Jure Leskovec. 2013. From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews. In *Proceedings of the 22nd International Conference on World Wide Web.*

[62] David D McDonald. 2010. Natural Language Generation. *Handbook of Natural Language Processing* (2010).

[63] 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.*

[64] Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating Style Transfer for Text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[65] Lingbo Mo, Chunhong Zhang, Yang Ji, and Zheng Hu. 2019. Adversarial Learning for Visual Storytelling with Sense Group Partition. In *Asian Conference on Computer Vision.*

[66] Aditya Mogadala, Marimuthu Kalimuthu, and Dietrich Klakow. 2020. Trends in Integration of Vision and Language Research: A Survey of Tasks, Datasets, and Methods. In *arXiv preprint arXiv:1907.09358.*

[67] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

[68] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems.*

[69] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

[70] Daniel Otter, Julian Medina, and Jugal Kalita. 2020. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[71] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

[72] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. 2019. Adversarial Inference for Multi-Sentence Video Description. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[73] Rakesh Patra and Sujan Kumar Saha. 2019. A Hybrid Approach for Automatic Generation of Named Entity Distractors for Multiple Choice Questions. *Education and Information Technologies* (2019).

[74] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *International Conference on Learning Representations.*

[75] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.*

[76] Rivindu Perera and Parma Nand. 2017. Recent Advances in Natural Language Generation: A Survey and Classification of the Empirical Literature. *Computing and Informatics* (2017).

[77] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[78] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *IEEE International Conference on Computer Vision.*

[79] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style Transfer Through Back-Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

[80] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020.*

[81] Lin Qiao, Jianhao Yan, Fandong Meng, Zhendong Yang, and Jie Zhou. 2020. A Sentiment-Controllable Topic-to-Essay Generator with Topic Knowledge Graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020.*

[82] Zhaopeng Qiu, Xian Wu, and Wei Fan. 2020. Automatic Distractor Generation for Multiple Choice Questions in Standard Tests. In *Proceedings of the 28th International Conference on Computational Linguistics.*

[83] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.*

[84] Sudha Rao and Joel Tetreault. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[85] Ehud Reiter and Robert Dale. 1997. Building Applied Natural Language Generation Systems. *Natural Language Engineering* (1997).

[86] Siyu Ren and Kenny Q. Zhu. 2021. Knowledge-Driven Distractor Generation for Cloze-Style Multiple Choice Questions. In *Thirty-Fifth AAAI Conference on Artificial Intelligence.*

[87] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-Critical Sequence Training for Image Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[88] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.*

[89] Yuri Safovich and Amos Azaria. 2020. Fiction Sentence Expansion and Enhancement via Focused Objective and Novelty Curve Sampling. In *IEEE 32nd International Conference on Tools with Artificial Intelligence.*

[90] Evan Sandhaus. 2008. The New York Times Annotated Corpus. LDC2008T19. Web Download. Philadelphia: Linguistic Data Consortium.

[91] Sashank Santhanam and Samira Shaikh. 2019. A Survey of Natural Language Generation Techniques with a Focus on Dialogue Systems - Past, Present and Future Directions. In *arXiv preprint arXiv:1906.00500.*

[92] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* (2009).

[93] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

[94] Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*

[95] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.*

[96] Min Shi, yufei tang, and Jianxun Liu. 2019. Functional and Contextual Attention-Based LSTM for Service Recommendation in Mashup Creation. *IEEE Transactions on Parallel and Distributed Systems* (2019).

[97] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision.*

[98] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.*

[99] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *Proceedings of the 36th International Conference on Machine Learning.*

[100] Milan J. Srinivas, Michelle M. Roy, and Viraj Kumar. 2019. Towards Generating Plausible Distractors for Code Comprehension Multiple-Choice Questions. In *2019 IEEE Tenth International Conference on Technology for Education.*

[101] Katherine Stasaski and Marti A. Hearst. 2017. Multiple Choice Question Generation Utilizing An Ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications.*

[102] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems.*

[103] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In *IEEE/CVF International Conference on Computer Vision.*

[104] Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and Position-aware Neural Question Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.*

[105] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems.*

[106] Jian Tang, Yue Wang, Kai Zheng, and Qiaozhu Mei. 2017. End-to-End Learning for Short Text Expansion. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[107] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems.*

[108] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations.*

[109] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. 2016. Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.*

[110] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[111] 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 *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.*

[112] Siyuan Wang, Zhongyu Wei, Zhihao Fan, Zengfeng Huang, Weijian Sun, Qi Zhang, and Xuanjing Huang. 2020. PathQG: Neural Question Generation from Facts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.*

[113] Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. 2019. A Multi-Agent Communication Framework for Question-Worthy Phrase Extraction and Question Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).

[114] Wei Wang, Hai-Tao Zheng, and Zibo Lin. 2020. Self-Attention and Retrieval Enhanced Neural Networks for Essay Generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing.*

[115] Yanmeng Wang, Wenge Rong, Jianfei Zhang, Yuanxin Ouyang, and Zhang Xiong. 2020. Knowledge Grounded Pre-Trained Model For Dialogue Response Generation. In *International Joint Conference on Neural Networks.*

[116] Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text.*

[117] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. 2016. What Value Do Explicit High Level Concepts Have in Vision to Language Problems?. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[118] Siying Wu, Zheng-Jun Zha, Zilei Wang, Houqiang Li, and Feng Wu. 2019. Densely Supervised Hierarchical Policy-Value Network for Image Paragraph Generation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.*

[119] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *IEEE Conference on Computer Vision and Pattern Recognition.*

[120] Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

[121] Ning Xu, Hanwang Zhang, An-An Liu, Weizhi Nie, Yuting Su, Jie Nie, and Yongdong Zhang. 2020. Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning. *IEEE Transactions on Multimedia* (2020).

[122] Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural Response Generation via GAN with an Approximate Embedding Layer. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.*

[123] Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing Topic-to-Essay Generation with External Commonsense Knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.*

[124] Pengcheng Yang, Fuli Luo, Peng Chen, Lei Li, Zhiyi Yin, Xiaodong He, and Xu Sun. 2019. Knowledgeable Storyteller: A Commonsense-Driven Generative Model for Visual Storytelling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.*

[125] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-Encoding Scene Graphs for Image Captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[126] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring Visual Relationship for Image Captioning. In *European Conference on Computer Vision.*

[127] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image Captioning with Semantic Attention. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[128] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2021. A Survey of Knowledge-Enhanced Text Generation. In *arXiv preprint arXiv:2010.04389*.

[129] Chenhan Yuan, Yi-Chin Huang, and Cheng-Hung Tsai. 2019. Efficient text generation of user-defined topic using generative adversarial networks. In *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*.

[130] Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine Comprehension by Text-to-Text Neural Question Generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

[131] 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*.

[132] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[133] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially Regularized Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*.

[134] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[135] Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017. Mechanism-Aware Neural Machine for Dialogue Response Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[136] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-End Dense Video Captioning with Masked Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[137] Linchao Zhu and Y. Yang. 2020. ActBERT: Learning Global-Local Video-Text Representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[138] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.