IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 9, 2020, accepted February 28, 2020, date of publication March 6, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979115

Generating Natural Language Descriptions From Tables

JUAN CAO, (Member, IEEE)
State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China
e-mail: caojuan@cuc.edu.cn

This work was supported in part by Fundamental Research Funds for Central Universities under Grant CUC200F010, and in part by the High-Quality and Cutting-Edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China).

ABSTRACT This paper proposes a neural generative architecture, namely NLDT, to generate a natural language short text describing a table which has formal structure and valuable information. Specifically, the architecture maps fields and values of a table to continuous vectors and then generates a natural language description by leveraging the semantics of a table. The NLDT architecture adopts a two-level neural model to make the most of the structure of a table to fully express the relationship between contents. To deal with the problem of out-of-vocabulary, this paper develops a simple and fast word-conversion method that replaces rare words appearing in texts with common field information in tables and directly replicates contents from table to the output sequence according to the field information. Besides, this paper adds the concept of theme to adapt the NLDT architecture to open domain and improves beam search algorithm to strengthen the results in the inference stage. On the WEATHERGOV dataset, the NLDT architecture improves the state-of-the-art BLEU-4 score from 61.01 to 62.89 and the current state-of-the-art F1 score from 73.21 to 78. On the WIKIBIO and WIKITABLE datasets, the NLDT architecture achieves a BLEU-4 score of 45.77 and 38.71 respectively which also outperform the state-of-the-art approaches. Furthermore, this paper introduces a Chinese dataset WIKIBIOCN including 33,244 biographies with corresponding tables. On the WIKIBIOCN dataset, the NLDT architecture achieves a BLEU-4 score of 38.87 and fairly good manual evaluation.

INDEX TERMS Table-to-text generation, NLDT, neural model, beam search, natural language generation, artificial intelligence.

I. INTRODUCTION

In general, tables have a formal structure that contains a set of records made up of fields (also known as properties) and values (also known as cells). Describing tables in natural language is very important in artificial intelligence to support some applications such as search engine and question answering system. Generating natural language descriptions from tables, also known as table-to-text generation, belongs to data-to-text generation in natural language generation. The aim is to help people better understand tables, especially those with complex data, such as weather data and medical data.

To achieve this task, the three main difficulties are how to reflect the structure of the table, how to express the relevance between the table and the text, and how to solve the rare words in the text. Different tables have different structures and characteristics. For example, each table in WEATHERGOV [1] consists of 36 predefined records in a fixed format, and words in the text rarely appear in the tables which are full of numbers and abbreviations. WIKIBIO [2] has a more complex structure, more types of fields and a very large vocabulary of text, totaling 400,000+ words. WIKITABLETEXT [3] is an open domain dataset containing 13,318 interpreted statements for 4962 tables. Therefore, a table representation method is needed to better express the contents of the table and reflect its characteristics. There are many contents in the table, but they appear very little in the text, so a good model is needed to express the relevance between the table and the text, and simulate the generation process from the table to the text. Some rare words often appearing in the text, such as name, region, etc. which usually also appear in the table, but not in the vocabulary whose size is limited that is called out-of-vocabulary. Copying words from the table to replace the unknown words in the output text is one way to solve the problem, but the copy may not be accurate enough.

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

This paper introduces an architecture generating natural language descriptions from tables (NLDT) to conquer the aforementioned difficulties. The proposed NLDT architecture consists of three parts: table preprocessing, two-level neural model and a word-conversion method. The purpose of table preprocessing is better expression of table content according to its characteristics, such as adding field types to numbers which may indicate temperature or rainfall. The two-level neural model based on seq2seq which is in an encoder-decoder framework is adopted to make full use of the table structure and express the relevance within tables and between tables and text. The encoder is responsible for transforming the processed table into continuous vector, and coding according to the content and structure of the table. The decoder is responsible for generating the text sequence according to the encoding result. Then using the word-conversion method designed by this paper locates and replaces the words in the output text that need to be replaced with the words in the table more accurately and quickly.

Besides, the table and text in different domains have different attributes and different text sequences respectively. So to adapt NLDT to open domain, the concept of theme is added to the model. In addition, the beam search algorithm is improved for gaining better results in the inference stage. Furthermore, this paper conducts and releases a Chinese dataset, WIKIBIOCN, and hope that it can offer opportunities to further research in this area, especially in Chinese.

Experiments are carried out on four datasets to verify the effectiveness of the NLDT architecture. On WIKIBIO and WIKITABLETEXT, NLDT improves the state-of-the-art BLEU-4 score from 44.89 to 45.77 and 38.23 to 38.71, respectively. On WEATHERGOV, NLDT improves the state-of-the-art BLEU-4 score from 61.01 to 62.89, and the state-of-the-art F1 score from 73.21 to 78.00. On the WIKIBIOCN, NLDT achieves the BLEU-4 scores of 38.87 and fairly good manual evaluation.

This paper is an extended version of our earlier works [4] and [5]. Reference [4] introduced a two-level model for table-to-text generation and [5] improved the model for open domain. This paper integrates the two with other methods to form a complete and relatively mature architecture, and greatly strengthens the interpretation and analysis. Specifically, this paper extends Section II for better comparison with previous work. This paper integrates the task of table-to-generation for single domain and open domain and adds a lot of method details in Section III and Section IV. In addition, this paper greatly strengthens the comparison and analysis of the experimental results in Section V. Moreover, this paper introduces a Chinese dataset WIKIBIOCN and conducts experiments on it with automatic evaluation and manual evaluation.

II. RELATED WORK

Without using neural network approaches, most works divide data-to-text generation task into two subtasks: content selection and surface realization. The purpose of content selection is to select relevant records from a table for discussion, whereas surface realization is to describe these records in natural language. These two subtasks can be accomplished with two separate modules. In terms of content selection, [6] proposed a method that allocates aligning records and sentences as a collective classification problem and [1] proposed a generative hierarchical semi-Markov method that aligns sequences with relevant records and then generates texts from these records. For surface realization, early works used templates [7] or statistically learning models, such as probabilistic context-free grammars [8] and language models [9], with hand-crafted features or rules. Generally, these works above are difficult in capturing the complexity of language.

In recent years, neural network plays an increasingly important role in NLG (Natural Language Generation). The neural encoder-decoder framework with attention mechanisms is widely used in many NLG tasks, such as neural machine translation [10] and video captioning [11], [12]. Recent progress has been made in using attention based encoder-decoder framework for table-to-text generation in which an input is encoded to a vector embedding, and then decoded to an output string of words using RNNs (Recurrent Neural Network). Although RNN is suitable to model variable length of text, including very long sentences, paragraphs and even documents [13], experiments on generating long texts from tables in [14] shows challenges in data-to-text generation and most of the work focuses on generating short texts from tables. Reference [15] proposed a seq2seq model with aligner between records and texts. The model is not capable to represent the tables with complex structure because it uses one-hot encoding as the structure is relatively simple. The approach of [2] is based on a conditional language model that is not suitable for long-range dependencies. Reference [16] introduced an order-planning text generation model to capture the relationship between different fields and used such relationship to make the generated text more fluent and smooth. Reference [17] trained a recurrent neural network seq2seq model with attention to select facts and generate textual summaries. Reference [18] used a fused bifocal attention mechanism which exploits and combines this micro and macro level information and a gated orthogonalization mechanism which tries to ensure that a field is remembered for a few time steps and then forgotten, and got different results on datasets in different languages such as English, French and German. Reference [19] proposed a neural generation system using a hidden semi-markov model (HSMM) decoder, which learns latent, discrete templates jointly with learning to generate and this model learns useful templates which make generation both more interpretable and controllable. Reference [20] interpreted the structured data as a corrupt representation of the desired output and used a denoising auto-encoder to reconstruct the sentence. Reference [21] proposed a neural network architecture equipped with copy actions that learns to generate single-sentence and comprehensible textual summaries from Wikidata triples which was mainly designed for open-domain

Wikipedia in different languages. A structure-aware seq2seq learning method in [22] and a table-aware seq2seq learning method in [3] are both taking into account the table structure and use copying mechanism to enhance the result, but [3] relies on a powerful copy mechanism to deal with the OOV problem and [22] ignores the effect of the fields. The work in this paper makes full use of the table structure and a fast word-conversion method help enhance copying mechanism greatly.

III. TASK FORMALIZATION

In this paper, the table-to-text generation is modeled as an encoder-decoder framework which takes a table as input and generates an intermediate description of the table and a word conversion process which converts the intermediate description to the final natural language output text.

The given table T can be viewed as a combination of n field-value records $\{R_1, R_2, \cdots, R_n\}$. Each record R_i consists of a field f_i and a sequence of values $\{d_1^i, d_2^i, \cdots, d_m^i\}$ and the field f_i is redefined as $\{f_1^i, f_2^i, \cdots, f_m^i\}$ for one-to-one correspondence with values. The table-to-text generation is formulated as an inference over a probabilistic model. The goal of the inference is to generate a sequence $u_{1:p}^*$ which maximizes $P(u_{1:p}|R_{1:n})$ as shown in (1). The intermediate generated description S' for table T which contains p tokens $\{u_1, u_2, \cdots, u_p\}$ with u_t being the intermediate output word at time t.

$$u_{1:p}^* = \arg\max_{u_{1:p}} \prod_{t=1}^{p} P\left(u_t | u_{0:t-1}, R_{1:n}\right) \tag{1}$$

There are three main types of the intermediate output words: field type, unknown word and vocabulary type, which individually identified by special markers. The final generated description S for table T which contains p tokens $\{w_1, w_2, \cdots, w_p\}$ with w_t being the final output word at time t. The corresponding relationship between w_t and u_t is shown in (2) which means w_t^* be transformed from u_t^* that if u_t^* is a field type and can be found in fields of the table such as a field f_j^k, w_t^* is set to the value d_j^k corresponding to the field f_j^k, or if u_t^* is a vocabulary type, w_t^* is set to u_t^* itself, or else w_t^* is copied from table according to u_t^*.

$$w_t^* = \begin{cases} d_j^k, & \text{if } u_t^* = f_j^k \\ u_t^*, & \text{if } u_t^* \in V_w \\ \text{copy}(u_t^*), & \text{others} \end{cases} \tag{2}$$

IV. ARCHITECTURE: NATURAL LANGUAGE DESCRIPTION FROM TABLE

A. NLDT

The overall diagram of NLDT is shown in Fig. 1. It includes embedding, encoding, decoding and word conversion.

The input of the NLDT is a table which includes a series of field-value pairs. Then after embedding, encoding and decoding, it outputs an intermediate text including three types of words. Finally, NLDT outputs the natural language text through word conversion from the intermediate text. Table 1 represents an example of input and output text in WIKIBIO using NLDT.

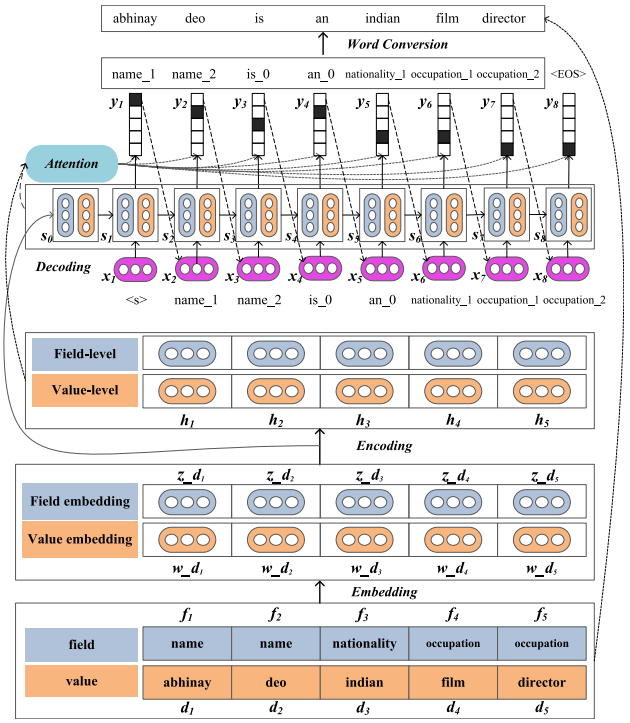

FIGURE 1. The overall diagram of NLDT.

TABLE 1. An example of input and output in WIKIBIO.

Field name	Value
name	abhinay
name	deo
nationality	indian
occupation	film
occupation	director
Intermediate Text	name_1 name_2 is_0 an_0 nationality_1 occupation_1 occupation_2
Final Output	abhinay deo is an Indian film director

1) EMBEDDING

A table can be thought of as a set of field-value records, in which the values are sequences or segments of words corresponding to certain fields. The representation of a table consists of word embedding, field embedding and position embedding. Word embedding w_d refers to words in the value content. Field embedding f_d embeds fields corresponding to words in the value content. Position embedding p_d refers to the position corresponding to the word in the value content. The position information is represented as a tuple (p_d^+, p_d^-) which includes the positions of the token d counted from the beginning and the end of the field respectively. Therefore, the united embedding as a key point to label each word in the field content is defined as a triple:

$$z\_d = \{f\_d; p\_d^+; p\_d^-\} \tag{3}$$

If one value always has only one word, the position information can be omitted. So p_d is not always necessary. When there is no position information, z_d is defined as f_d.

In the training stage, word embedding, field embedding and position embedding are all randomly initialized.

2) ENCODER

In this paper, the relations between words and fields, between words and between fields are all considered to be very important, so NLDT adopts a two-level LSTM-RNN (Long Short-Term Memory, Recurrent Neural Network) encoder. LSTM-RNN uses a vector of cell state and a set of element-wise multiplication gates to control how information is stored, forgotten and exploited inside the network. Following the design of an LSTM cell in [23], the two-level encoder is defined by following equations that specifically the field-level encoder is defined from (4) to (7), and the value-level encoder is defined from (8) to (11).

$$\begin{pmatrix} i_t^z \\ f_t^z \\ o_t^z \\ \hat{c}_t^z \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \tanh \end{pmatrix} W_{4n,2n}^z \begin{pmatrix} z\_d_t \\ h_{t-1}^z \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} l_t^z \\ \hat{w\_d_t} \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \tanh \end{pmatrix} W_{2n,n}^z (w\_d_t) \tag{5}$$

$$c_t^z = f_t^z \odot c_{t-1}^z + i_t^z \odot \hat{c}_t^z + l_t^z \odot \hat{w\_d_t} \tag{6}$$

$$h_t^z = o_t^z \odot \tanh(c_t^z) \tag{7}$$

At the field level, the encoder mainly focuses on the influence between fields and the influence of values on fields, so it encodes each united embedding z_d_j together with its word embedding w_d_j into the hidden state h_j^z.

$$\begin{pmatrix} i_t^d \\ f_t^d \\ o_t^d \\ \hat{c}_t^d \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \text{sigmoid} \\ \text{sigmoid} \\ \tanh \end{pmatrix} W_{4n,2n}^d \begin{pmatrix} w\_d_t \\ h_{t-1}^d \end{pmatrix} \tag{8}$$

$$\begin{pmatrix} l_t^d \\ \hat{f\_d_t} \end{pmatrix} = \begin{pmatrix} \text{sigmoid} \\ \tanh \end{pmatrix} W_{2n,n}^d (f\_d_t) \tag{9}$$

$$c_t^d = f_t^d \odot c_{t-1}^d + i_t^d \odot \hat{c}_t^d + l_t^d \odot \hat{f\_d_t} \tag{10}$$

$$h_t^d = o_t^d \odot \tanh(c_t^d) \tag{11}$$

At the value level, the encoder mainly focuses on the influence between values and the influence of fields on values, so it encodes each word embedding z_d_j together with its field embedding f_d_j into the hidden state h_j^d.

3) DECODER

In decoder, a two-level LSTM-RNN with attention is used to generate the intermediate description. As defined in the (2), the generated token u_t at time t in the decoder is predicted based on all previously generated tokens $u_{<t}$ before u_t, and the two-level encoder hidden states $H^z = \{h_t^z\}_{t=1}^L$ and $H^d = \{h_t^d\}_{t=1}^L$. The two-level decoder includes a field-level decoder and a value-level decoder. The field-level decoder uses attention a_t^z to compute the relevance between the field-level encoder states H^z and the field-level decoder state s_t^z. The word-level decoder uses attention a_t^d to compute the relevance between the word-level encoder states H^d and the word decoder state s_t^d at time t. To be more specific:

$$P(u_t|H^z, H^d, u_{<t}) = \text{softmax}(W_s[g_t^z, g_t^d]) \tag{12}$$

$$g_t^z = \tanh(W_t^z[s_t^z, a_t^z]) \tag{13}$$

$$g_t^d = \tanh(W_t^d[s_t^d, a_t^d]) \tag{14}$$

$$s_t^z = \text{LSTM}(u_{t-1}, s_{t-1}^z) \tag{15}$$

$$s_t^d = \text{LSTM}(u_{t-1}, s_{t-1}^d) \tag{16}$$

$$a_{t_i}^z = \frac{e^{g(s_t^z, h_i^z)}}{\sum_{j=1}^N e^{g(s_t^z, h_j^z)}}; a_t^z = \sum_{i=1}^L a_{t_i}^z h_i^z \tag{17}$$

$$a_{t_i}^d = \frac{e^{g(s_t^d, h_i^d)}}{\sum_{j=1}^N e^{g(s_t^d, h_j^d)}}; a_t^d = \sum_{i=1}^L a_{t_i}^d h_i^d \tag{18}$$

$$g(s_t^z, h_i^z) = \tanh(W_p^z h_i^z) \odot \tanh(W_q^z s_t^z) \tag{19}$$

$$g(s_t^d, h_i^d) = \tanh(W_p^d h_i^d) \odot \tanh(W_q^d s_t^d) \tag{20}$$

According to (19) and (20), dot product is used to measure the similarity between s_t^z and h_i^z, and between s_t^d and h_i^d. W_s, W_t^z, W_t^d, W_p^z, W_q^z, W_p^d and W_q^d are all model parameters.

4) WORD CONVERSION

Words in a text always come from the corresponding table. Most of these words appear infrequently in other texts, but their fields more likely occur frequently. This paper makes use of the field information to replace the rare words in texts to improve the accuracy of generation. In the training stage, if the words in texts also appear in the value contents of their corresponding tables, their field information and position information form new words of field type which is one kind of words in intermediate texts. The words not appearing in corresponding tables are marked with a special symbol as vocabulary type in intermediate texts. Taking WIKIBIO as an example, if the word in the text appears in the table, its field name and position of occurrence in the value content are recorded, and then the word w is replaced with a new word u consisting of the field name plus the position (e.g., name_1). If the word w is not in the table, a number like zero is added to the back of the word as a new word u (e.g., born_0). These new words form a new text S' used by the decoder for training. The vocabulary constituted by new words of texts is the target vocabulary in decoder.

The last step is word conversion from the intermediate output in decoder. The generated tokens in the intermediate output can be divided into three kinds. One kind ends with zero representing it is not in the table, so the final word is just need to be extracted part from it (e.g., born_0 -> born). The second kind ends with nonzero meaning it appears in the table, so the field name and position are extracted from it, then searching the word in the table based on its field and position (e.g., name_1 -> name and 1 -> John). If not found, the word is treated as an unknown token (UNK) which is the

third kind. UNK is replaced with the most relevant token in the corresponding table according to the related field-level attention matrix.

After word conversion, The final natural language text is obtained. Through reprocessing text in the training and word conversion, it not only accelerates the convergence speed of training, but also alleviates the OOV problem greatly.

B. NLDT +

In order to adapt to open domain, this paper designs NLDT+ by adding the theme concept to NLDT. A theme is the center of a text or the category of objects that are described, such as characters, television dramas, competitions or nationalities. If the theme is not explicitly given in the table, it is difficult to know. So how is the theme of the text determined? Firstly, the text of different domains has different word sequences. In addition, the table fields corresponding to different domains are also very different, such as the date of birth of the characters, the date of broadcast of the TV series, the score of the competition, and so on. In this paper, the theme is represented by the encoded result which becomes the initial state of the decoder and is added to the decoding process, specifically changing (13) and (14) to (21) and (22).

$$g_t^z = \tanh(W_t^z[s_t^z, a_t^z, s_0^z]) \tag{21}$$

$$g_t^d = \tanh(W_t^d[s_t^d, a_t^d, s_0^d]) \tag{22}$$

C. BEAM SEARCH

In the inference stage, beam search is used to find better results. The width of beam search k represents the number of candidate results. Theoretically, the larger k is, the better the result is, but the computational complexity will also increase. Because the length of the sequence will affect the results of beam search, length normalization is usually used. In the experiment, it is not good to choose the first candidate as the final result directly. In the process of beam search, rather than the length, the probability of words is a more important impact factor for which word to be chosen next. In the end of beam search, length needs to be taken into account to pick the best one sentence from the candidates and the influence of length is greater. In order to reduce the influence of length, this paper introduces two-stage length normalization in the process and at the end of beam search, respectively. Sometimes, the coverage of table content in text content is also very important which means how much table information a text can described. Therefore, combining the two aspects forms an improved algorithm. The equations are as follows:

$$S(Y|X) = \frac{lp + cp}{|Y|^{\gamma}} \tag{23}$$

$$lp = \frac{P(Y|X)}{|Y|^{\alpha_2}} \tag{24}$$

$$cp = \beta * \frac{\sum_{i=1}^{|X|} \log\left(\min(\sum_{j=1}^{|Y|} a_{i,j}^z, 1.0) + \min(\sum_{j=1}^{|Y|} a_{i,j}^d, 1.0)\right)}{|X| * |Y|} \tag{25}$$

$$P(Y|X) = \arg\max \frac{1}{T_y^{a_1}} \sum_{t=1}^{T_y} \log P(y^{<t>}|x, y^{<1>}, \cdots, y^{<t-1>}) \tag{26}$$

The two-stage length normalization occurs in beam search process and at the end of beam search, respectively. In the process of beam search, the length normalization is defined in (26) that α_1 is the corresponding length normalization factor and $P(Y|X)$ is the probability obtained by beam search with the first-stage length normalization. If α_1 is larger, the influence of length on probability is smaller. The second-stage length normalization is defined in (24) that α_2 is the corresponding length normalization factor and lp represents the result of the second-stage length normalization to the probability of candidate result in the end of beam search. If $\alpha_1 < \alpha_2$, the effect of length during beam search will be larger than that in the end of beam search which is also proven to yield better results. In (25), cp is defined as the extent of the coverage of table content in text content that β is a factor deciding the importance of the coverage which is calculated from attention result $a_{i,j}^z$ at the field level and $a_{i,j}^d$ at the value level during decoding. In (23), $S(Y|X)$ represents the final score combining the two parts lp and cp. Then the candidate result with the highest score is picked as the final output. This paper uses greedy search in the training and beam search in the testing.

TABLE 2. Characteristics of four datasets.

	WIKIBIO	WIKITABLETEXT	WEATHERGOV	WIKIBIOCN
Language	English	English	English	**Chinese**
Domain	Biography	**Open domain**	Weather	Biography
Main types of Structured data	Non-numeric text	Non-numeric text	**Numbers and Symbols**	Non-numeric text
Characteristic	**Large vocabulary and long sentences**	Different attributes in different domains and more rare words	Difficulty in choosing the right number	Chinese word segmentation and long sentences

V. EXPERIMENTS

This section presents experiments on four datasets: WEATHERGOV, WIKIBIO, WIKITABLETEXT, and WIKIBIOCN. These four datasets have their own characteristics as shown in Table 2.

A. EXPERIMENT ON WEATHERGOV

1) DATASET

WEATHERGOV introduced by [1] consists of 29,528 scenarios, each with a table containing corresponding 36 records (e.g., temperature, wind direction etc.) paired with a natural language forecasts (28.7 avg. word length) for 3,753 cities in the U.S.A over three days. The values of table are either

TABLE 3. Parameter settings on WEATHERGOV.

Word dimension	Field dimension	Hidden size	Batch size
100	50	400	16

TABLE 4. BLEU-4 and F1 on WEATHERGOV.

Model	BLEU-4	F1
KL	36.45	-
AKL	38.40	65.4
MBW	61.01	73.21
NLDT	**62.89**	**78.00**

numbers or pre-defined categories and the records contain 12 types. The experiment uses WEATHERGOV training, development, and test splits of size 25,000, 1,000 and 3,528, respectively.

2) SETTING

Due to the vocabulary of this dataset is totally less than 400 words, all words and fields from the table of the training set are taken as the source word vocabulary and the field vocabulary respectively in encoder, and all reprocessed words from the sentence of the training set are taken as the target word vocabulary in decoder.

In addition, every value has only one word, so the number of position is set 0 or 1, and every value in table is added with its record type (e.g. temperature_30). The initial learning rate is 0.001 with using adam optimizer. This experiment adopts greedy search in the inference stage. The other model parameters are displayed in Table 3.

3) RESULT

This experiment has gone through sixteen epochs altogether. The generation quality is assessed automatically with BLEU-4 and F1 measure. Table 4 compares the test result against previous methods that include two non-neural network models KL [24] and AKL [9], and a neural network model MBW [15] that is an improved attention model with additional regularized terms.

The values of the table in WEATHERGOV are mostly numerical values or fixed patterns like 'S' and 'SC'. The numbers under different environments such as 'temperature' and 'windSpeed' have different meanings. So it is difficult to put all numbers of all types into the vocabulary. But through the word conversion, the required values can be accurately found in the table by NLDT after a large quantity of training. A sample of table, reference text and generated text is shown in Fig. 2 that the number '23' is corresponding to the max number of gust.

temperature(time=17-30, min=52, mean=57, max=67)
windSpeed(time=17-30, min=10, mean=12, max=16)
windDir(time=17-30, mode=S)
gust(time=17-30, min=0, mean=7, max=23)
skyCover(time=17-30, mode=75-100)
skyCover(time=17-21, mode=50-75)
precipChance(time=17-30,min=12,mean=12,max=12)

(R) mostly cloudy , with a low around 50 . south wind between 10 and 15 mph , with gusts as high as 20 mph .
(G) mostly cloudy , with a low around 50 . south wind between 10 and 15 mph , with gusts as high as 23 mph .

FIGURE 2. Sample table and reference text (R) chosen from WEATHERGOV and corresponding generated text (G) by NLDT.

TABLE 5. Parameter settings on WIKIBIO.

Word dimension	Field dimension	Position dimension	Hidden size	Batch size
400	90	6	500	16

B. EXPERIMENT ON WIKIBIO

1) DATASET

WIKIBIO is introduced by [2] for generating biography from an infobox. An infobox can be viewed as a table with a set of field-value records. WIKIBIO contains 728,321 articles from English Wikipedia (Sep 2015). The dataset extracts the first sentence of each biography article as reference. On average, every sentence has 26.1 tokens of which 9.5 tokens occur in the table and every table has 53.1 tokens and 19.7 fields. The corpus has been divided into training (80%), testing (10%) and validation (10%) sets.

2) SETTING

In encoder, 15,093 words occurring more than 100 times and 3,004 fields occurring more than 10 times from the table of the training set are selected as the source word vocabulary and the field vocabulary respectively. In decoder, the most frequent 20,000 reprocessed words from the sentence of the training set are selected as the target word vocabulary.

In addition, empty fields are filtered and the position number is restricted to 50. The initial learning rate is 0.001 with using adam optimizer. In the inference stage, this experiment adopts greedy search in training and beam search (two-stage length normalization) with the beam size $k = 6$ and the length penalty $\alpha_1 = 0.5$ and $\alpha_2 = 1.0$ in testing. The other model parameters are displayed in Table 5.

3) RESULT

This experiment has gone through five epochs altogether. The generation quality is assessed automatically with BLEU-4. Table 6 shows the results of comparisons of various models on WIKIBIO.

Table NLM introduced by [2] includes local and global conditioning over the table by integrating related field and position embedding into the table representation.

MBW introduced by [15] uses an improved attention model with additional regularized terms which influence the weights assigned to the fields.

TABLE 6. BLEU-4 on WIKIBIO.

Model	BLEU-4
Table NLM	34.70
MBW	35.10
Basic Seq2seq	38.20
Table2Seq	40.26
Table2Seq w/o Copying	36.88
BAMGO	42.03
Vanilla Seq2Seq	43.65
Structure-aware Seq2Seq	44.89
NLDT	44.53
+ Copying	44.70
+ Beam Search	**45.77**

Basic Seq2seq introduced by [25] is a simple basic encode-decode model with a copying mechanism.

Table2Seq introduced by [3] is a neural generative model that maps a table to continuous vectors and then generates a natural language sentence by leveraging the semantics of a table with a flexible copying mechanism.

BAMGO introduced by [18] used a fused bifocal attention mechanism which exploits and combines this micro and macro level information and a gated orthogonalization mechanism which tries to ensure that a field is remembered for a few time steps and then forgotten.

Vanilla seq2seq neural architecture introduced by [22] uses the concatenation of word embedding, field embedding and position embedding as the model input. The model can operate local addressing over the table by the natural advantages of LSTM units and word level attention mechanism.

Structure-aware Seq2Seq introduced by [22] consists of field-gating encoder and description generator with dual attention to encode both the content and the structure of a table.

This paper also compares the results of NLDT with greedy search or beam search and with copy or no-copy. The role of copy mechanism is not obvious for NLDT. The main reason is that the model adopts a word-conversion method, which is equivalent to a powerful copy. But to Table2Seq, it differs greatly in the use of copy mechanism and in the absence of copy mechanism. In the inference stage, this paper adopts beam search with two-stage length normalization that further improve the results on WIKIBIO.

Table 7 shows some examples of generated sentences by NLDT on WIKIBIO. While NLDT could generate fluent and table related sentences, some problems can be found by comparing with reference texts such as lack of information, reversal of sentence sequence, lack of abbreviations for words and so on.

The process of text generation is visually displayed in Fig. 3 that attention is used to choose the content of the table. For example, 'film' in the table is focused by the attention when generating 'occupation_1' in the intermediate output corresponding to 'film' in the final output.

TABLE 7. Examples of the reference and generated sentences by NLDT on WIKIBIO.

Group	Text (R: Reference G: Generated)
(1)	(R) gilbert joseph `` bus '' griffiths ( 1913 -- september 25 , 2006 ) was **a cartoonist , lumberjack , and fisherman** . (G) gilbert joseph `` bus '' griffiths (1913 -- september 25 , 2006) was **a canadian cartoonist** .
(2)	(R) lonnie kimble (born january 3 , 1990) is an american rapper , better known by his stage name skeme . (G) lonnie kimble (born january 3 , 1990) , better known by his stage name skeme , **is an american rapper** .
(3)	(R) jan kubice (**b.** 3 october 1953) is a czech politician , who served as minister of the interior **of the czech republic** from **april** 2011 to **july** 2013 . (G) jan kubice (**born** 3 october 1953) is a czech politician , who served as minister of the interior from 2011 to 2013 .

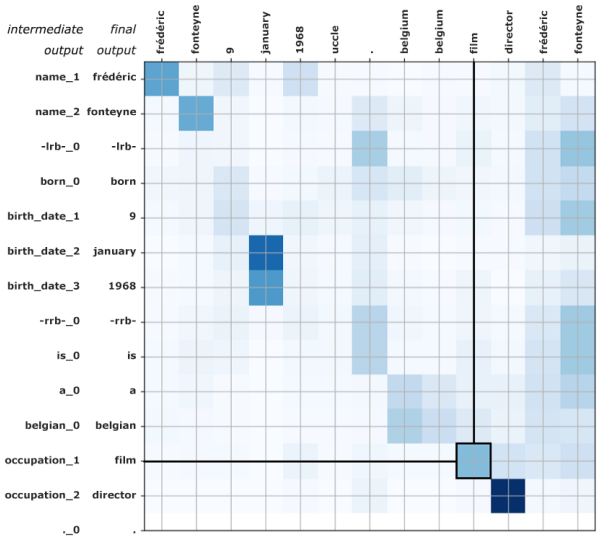

FIGURE 3. Attention on WIKIBIO in intermediate and final output.

Through model training, a subsidiary product is also obtained, that is, word vectors related to vocabulary. The vocabulary vectors are visualized and displayed in three-dimensional space after dimensionality reduction by principal component analysis (PCA), as shown in Fig. 4. By searching for a word like 'musician', the list of words closest to it appears such as 'jazz', 'singer', 'dj', 'songwriter' and so on.

C. EXPERIMENT ON WIKITABLETEXT

1) DATASET

WIKITABLETEXT is the first open-domain dataset for table-to-text generation introduced by [3]. Its tables were crawled from Wikipedia, and texts were annotated by manpower. There are 13,318 row-text pairs in WIKITABLETEXT that is divided into training (10,000), development (1,318), and test (2,000) sets.

2) SETTING

In encoder, 4,155 words occurring more than 2 times and 2,423 attributes occurring more than 1 times from the table of

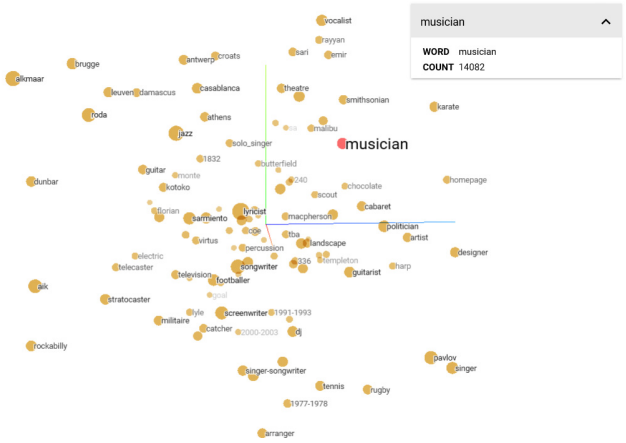

FIGURE 4. Word vectors related to vocabulary on WIKIBIO.

TABLE 8. Parameter settings on WIKITABLETEXT.

Word dimension	Field dimension	Hidden size	Batch size
200	100	300	16

the training set are selected as the source word vocabulary and the attribute vocabulary respectively. In decoder, 3,329 reprocessed words occurring more than 1 times from the sentence of the training set are selected as the target word vocabulary and theme concept is added to the decoder.

The initial learning rate is 0.001 with using adam optimizer. In the inference process of testing, the improved beam search with two-stage length normalization and the coverage of table content in text content introduced by this paper is used with $\alpha_1 = 0.3$, $\alpha_2 = 0.4$, $\beta = 0.7$, $\gamma = 0.8$ and the beam size $k = 7$. The other parameters are displayed in Table 8.

3) RESULT

This experiment has gone through eight epochs altogether. Table 9 compares the results of BLUE-4 by NLDT+ in this paper against that by the previous methods including ''TC-NLM'', ''Random-Copying'', ''Template'' and ''Table2seq'' all introduced by [3].

TC-NLM is a table conditional neural language model baseline, which is based on a recurrent neural network language model introduced by [26] and the model is fed with local and global factors to make it also consider table information.

Random-Copying does not use the copying mechanism but replaces the <unk> with a randomly selected cell from the table.

Template is constructed with a list of templates ranked by the frequency for each table scheme. The templates are derived by replacing the cells appearing in the text with the corresponding fields. In the inference process, the template with the highest frequency given a table scheme is selected,

TABLE 9. BLEU-4 on WIKITABLETEXT.

Model	BLEU-4
TC-NLM	5.79
Random-Copying	12.01
Template	28.62
Table2Seq	38.23
w/o Copying	5.41
NLDT	35.14
NLDT+	**38.71**
w/o Copying	37.24
Beam Search (β=0)	38.47
Beam Search (α_2=0)	38.25
Beam Search (α_2=0, β=0)	37.93
Beam Search (α_2=0, β=0, γ=0)	35.98
Disordered Table Fields	37.54

and the fields with the corresponding values are replaced to generate text.

Table2seq mainly uses global and local information in its encoder-decoder architecture and a powerful copying mechanism to solve the OOV problem. Table2seq also uses beam search in the inference process.

Theme. The results show that NLDT is far less effective than NLDT+. Because NLDT+ adds the theme concept to NLDT. So the theme concept is proved useful to the ability to generate for open domain.

Copying. Since words from table cannot be well covered in the vocabulary especially for open domain, they could not be generated in the predicted sequence. Enlarging vocabulary size is not a principal way to solve this problem because it could not handle the rare words not seen in the training data. Removing the copying mechanism from Table2seq gets down its performance from 37.90 to 5.41. The performance of randomly copying is also poor. It is reasonable because many words of the description come from table cells which are rare words and need to be copied. So copying mechanism is the most effective factor to Table2seq.

But the results show removing the copying mechanism from NLDT+ does not dramatically decrease performance. The word-conversion method in this paper is equivalent to a powerful copy. Most of rare words are replaced with the corresponding fields information which greatly reduces the number of rare words and the vocabulary is also reduced a lot.

Beam Search. The improved beam search adopts two-stage length normalization and considers the coverage of table content in text content. In order to test the effectiveness of this method, this experiment sets the relevant parameters to zero. The second-stage length normalization is turned into invalid through $\alpha_2 = 0$, and the coverage of table content in text content becomes unaffected by $\beta = 0$. The original

TABLE 10. Examples of the reference and generated sentences by NLDT on WIKITABLETEXT.

Group	Text (R: Reference G: Generated)
(1)	(R) viktor gerashchenko is the chairman of central bank of russia during **1992 to 1994** . (G) viktor gerashchenko is the chairman of central bank of russia during **1992—1994** .
(2)	(R) greatest hits live is a **discography** of ace frehlely in 2006 . (G) ace frehley **released** the greatest hits live in 2006 .
(3)	(R) american regions mathematics league was **held** at brown university in 1979 . (G) american regions mathematics league was **published** in brown university in 1979 .

TABLE 11. The most common 11 fields in WIKIBIOCN.

Field name	Occurrences
page_title	30000
name	22612
birth_date	18143
image	16999
birth_place	16724
nationality	8267
death_date	7040
occupation	6935
caption	6637
position	6340
death_place	6026

beam search with length normalization is equivalent to the improved beam search when $\alpha_2 = 0$, $\beta = 0$ and $\gamma = 0$. The results show that the improved beam search introduced in this paper is better than the original one and both sides of the improved method are helpful to improve the effect.

Disordered Tables. A structured table consists of a set of field-value records and the records are fed into the generator sequentially as the order they are presented in the table. Reference [27] points out the order of records can guide the description generator to produce an introduction in the predefined schemas. However, not all the tables are arranged in the proper order. Furthermore, the schemas of various types of tables differ greatly from each other. For these reasons, this paper disorders tables in WIKITABLETEXT by randomly shuffling the records of a table to test the performance of NLDT. The results show that although NLDT performs not as good as before which means the order of table records is an essential aspect for table-to-text generation, the decrease in BLEU-4 is very small which means NLDT can mitigate this impact of disordered tables.

Table 10 shows some examples of generated sentences of NLDT on WIKITABLETEXT. These generated sentences could basically express the same meaning as the reference text sentences with different ways of expression, but some problems remain to be solved such as improper choice of words, unable to split words representing time periods and so on.

D. EXPERIMENT ON WIKIBIOCN

1) DATASET

For researchers, the use of International Open English datasets is convenient to compare with previous work to illustrate the quality of the models. English spaces are used as separators that lexical selection and text comparison will not be affected by word segmentation. Some researchers have studied the influence of different languages on models such as German, French, Arabic, Esperanto, and so on. Since there is no Chinese dataset that can be used for table-to-text generation, this paper chooses Chinese Wikipedia as data source and sorts out a Chinese dataset WIKIBIOCN [28], in order to verify the feasibility of the methods introduced by this paper and provide a reference for other researchers.

TABLE 12. The number of field types and vocabulary in WIKIBIOCN.

	Field types	Vocabulary
training set	4794	167146
test set	1295	17889
verification set	1679	30757

TABLE 13. An example of table and corresponding text in WIKIBIOCN.

Field name	Value	Field name	Value
name	范增	sex	男
image file	范增像.jpg	birth_date	前 278 年
occupation	政治人物	birth_place	楚国居巢
period	前 3 世纪	death_date	前 204 年
nationality	楚	death_place	彭城附近
past	楚谋士	work	鸿门宴项庄舞剑
page_title	范增		

Text: 范增（前 278 年－前 204 年），战国后期至秦末居巢（今安徽省巢湖市亚父街道）人，西楚霸王项羽首席谋臣幕僚。

WIKIBIOCN includes 33,244 biography sentences with related tables which come from Chinese Wikipedia (July 2018). Most of Chinese characters in Chinese Wikipedia are traditional characters, so it first needs to convert them to simplified characters. Different from English dataset like WIKIBIO, Chinese character is the basic unit in Chinese but the word is actually more semantic, so it must undergo word segmentation which may also affect the result. After simplified transformation and word segmentation, the sentence has 248 words at the longest, 12 words at the shortest, and 43.9 words on average. The number of attributes in one table is 60 at most and 6 at least. The dataset is divided into training set (30,000), verification set (1000) and test set (2,244).

Table 11 represents the most common 11 fields and occurrences in the training set. Table 12 represents the number of field types and vocabulary in training set, test set and verification set. Table 13 represents an example of table and corresponding text in WIKIBIOCN.

2) SETTING

In encoder, words occurring more than 9 times and fields occurring more than 1 times from the table of the training set are selected as the source word vocabulary and the field vocabulary respectively. In decoder, the reprocessed words occurring more than 2 times from the sentence of the training

TABLE 14. Parameter settings on WIKIBIOCN.

Word dimension	Field dimension	Position dimension	Hidden size	Batch size
400	100	10	500	16

TABLE 15. BLEU-4 on WIKIBIOCN.

	BLEU-4
NLDT	38.20
w/o Copying	37.50
NLDT + Beam Search	**38.87**

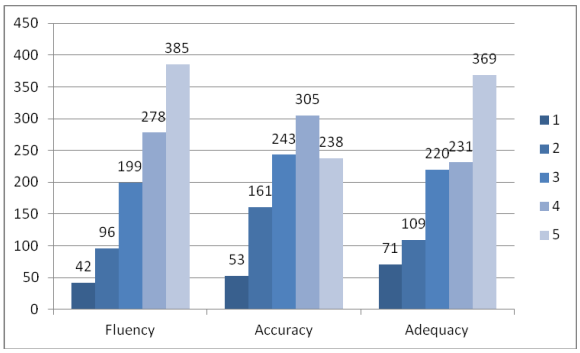

FIGURE 5. Subjective score of test result on WIKIBIOCN.

set are selected as the target word vocabulary. The initial learning rate is 0.005 with using adam optimizer. In the inference process of testing, beam search (two-stage length normalization) is used with the length penalty $\alpha_1 = 1.0$, $\alpha_2 = 1.0$ and the beam size $k = 7$. The other model parameters are displayed in Table 14.

3) RESULT

The generation quality is assessed automatically with BLEU-4. Table 15 shows the results of experiments on WIKIBIOCN. The results show that the NLDT can also be applied to the table-to-text generation in Chinese, and the use of beam search algorithm can help the model to generate text better.

100 generated texts are randomly selected from the test results, and are graded by 10 public judges from three perspectives: fluency, accuracy and adequacy. The statistical results are shown in Fig. 5. Accuracy reflects the consistency of expression in reference text and generated text, which is a very important metric. Adequacy reflects whether the generated text contains all the content expressed by the reference text. The results show that the fluency and adequacy of the generated text are affirmed by the public, and the accuracy is relatively low. Most of the texts are fluent, and a few of them have textual structure problems. The main reasons for the inconvenience are sentence duplication and word selection errors.

VI. CONCLUSION

This paper presents a neural generative architecture NLDT for table-to-text generation and conducts experiments on WEATHERGOV, WIKIBIO, WIKITABLETEXT and WIKIBIOCN which demonstrates the effectiveness of this approach. In future, this work will combine knowledge graph to enhance the comprehension and realize personalized text generation based on user goals.

REFERENCES

[1] P. Liang, M. I. Jordan, and D. Klein, "Learning semantic correspondences with less supervision," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (AFNLP-ACL-IJCNLP)*, 2009, pp. 91–99.

[2] R. Lebret, D. Grangier, and M. Auli, "Neural text generation from structured data with application to the biography domain," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2016, pp. 1203–1213.

[3] J. Bao, D. Tang, N. Duan, Z. Yan, M. Zhou, and T. Zhao, "Text generation from tables," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 27, no. 2, pp. 311–320, Feb. 2019.

[4] J. Cao, J. Gong, and P. Zhang, "Two-level model for table-to-text generation," in *Proc. SSPS*, New York, NY, USA, 2019, pp. 121–124.

[5] J. Cao, J. Gong, and P. Zhang, "Open-domain table-to-text generation based on Seq2seq," in *Proc. Int. Conf. Algorithms, Comput. Artif. Intell. (ACAI)*, New York, NY, USA, 2018, pp. 1–5.

[6] R. Barzilay and M. Lapata, "Collective content selection for concept-to-text generation," in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT/EMNLP)*, 2005, pp. 331–338.

[7] K. Van Deemter, M. Theune, and E. Krahmer, "Real versus template-based natural language generation: A false opposition?" *Comput. Linguistics*, vol. 31, no. 1, pp. 15–24, Mar. 2005.

[8] A. Belz, "Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models," *Natural Lang. Eng.*, vol. 14, no. 4, pp. 431–455, Oct. 2008.

[9] G. Angeli, P. Liang, and D. Klein, "A simple domain-independent probabilistic approach to generation," in *Proc. EMNLP*, 2010, pp. 502–512.

[10] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang, and J. Xie, "A hierarchy-to-sequence attentional neural machine translation model," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 623–632, Mar. 2018.

[11] J. Song, Y. Guo, L. Gao, X. Li, A. Hanjalic, and H. T. Shen, "From deterministic to generative: Multimodal stochastic RNNs for video captioning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3047–3058, Oct. 2019.

[12] L. Gao, X. Li, J. Song, and H. T. Shen, "Hierarchical LSTMs with adaptive attention for visual captioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[13] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. EMNLP*, 2015, pp. 1422–1432.

[14] S. Wiseman, S. Shieber, and A. Rush, "Challenges in data-to-document generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2017, pp. 2253–2263.

[15] H. Mei, M. Bansal, and M. R. Walter, "What to talk about and how? Selective generation using LSTMs with Coarse-to-Fine alignment," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 720–730.

[16] L. Sha, L. Mou, and T. Liu, P. Poupart, S. Li, B. Chang, and Z. Sui, "Order-planning neural text generation from structured data," in Proc. AAAI, 2018, pp. 1–8.

[17] A. Chisholm, W. Radford, and B. Hachey, "Learning to generate one-sentence biographies from Wikidata," 2017, *arXiv:1702.06235*. [Online]. Available: https://arxiv.org/abs/1702.06235

[18] P. Nema, S. Shetty, P. Jain, A. Laha, K. Sankaranarayanan, and M. M. Khapra, "Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., (Long Papers)*, 2018, pp. 1539–1550.

[19] S. Wiseman, S. M. Shieber, and A. M. Rush, "Learning neural templates for text generation," 2018, *arXiv:1808.10122*. [Online]. Available: https://arxiv.org/abs/1808.10122

[20] M. Freitag and S. Roy, "Unsupervised natural language generation with denoising autoencoders," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3922–3929.
[21] L.-A. Kaffee, H. Elsahar, P. Vougiouklis, C. Gravier, F. Laforest, J. Hare, and E. Simperl, "Learning to generate wikipedia summaries for underserved languages from wikidata," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Short Papers)*, 2018, pp. 640–645.
[22] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui, "Table-to-text generation by structure-aware seq2seq learning," in *Proc. AAAI*, 2018, pp. 4881–4888.
[23] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[24] I. Konstas and M. Lapata, "Inducing document plans for concept-to-text generation," in *Proc. EMNLP*, 2013, pp. 1503–1514.
[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Comput. Sci.* to be published.
[26] T. Mikolov, "Recurrent neural network based language model," in *Proc. ISCA*, 2010, pp. 1045–1048.
[27] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," 2015, *arXiv:1511.06391*. [Online]. Available: https://arxiv.org/abs/1511.06391
[28] J. Cao, "WIKIBIOCN," *IEEE Dataport*, to be published. Accessed: Jan. 9, 2020, doi: 10.21227/06v1-tx61.

JUAN CAO (Member, IEEE) was born in Datong, China, in 1990. She received the B.S. degree in computer science and technology, the M.S. degree in computer application technology, and the Ph.D. degree in new media from the Communication University of China, in 2012, 2015, and 2019, respectively.

Since July 2019, she has been a Postdoctoral Research Associate with the Collaborative Innovation Center, Communication University of China. She is currently a member of the State Key Laboratory of Media Convergence and Communication, Communication University of China. Her research interests include data-to-text generation with deep learning, knowledge graph, and applications of robotic journalism in media fusion.

Dr. Cao is a member of the China Computer Federation (CCF). She won the National Scholarship for Postgraduates, in 2015.