doi: 10.1093/comjnl/bxaa048

Encoder–Decoder Couplet Generation Model Based on 'Trapezoidal Context' Character Vector

Rui Gao[1], Yuanyuan Zhu[2], Mingye Li[3], Shoufeng Li[1] and Xiaohu Shi[1,*]

[1]*Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University, 2699 Qianjin Street, Changchun 130012, China*

[2] *JD Cloud & AI Department, Beijing Wodong Tianjun Information Technology Co., Ltd. 18th House, Kechuang 11th Street, Daxing District, Beijing 101111, China*

[3]*School of Computing and Information Systems, Melbourne School of Engineering, The University of Melbourne, Victoria 3010, Australia*

*****Corresponding author: shixh@jlu.edu.cn*

This paper studies the couplet generation model which automatically generates the second line of a couplet by giving the first line. Unlike other sequence generation problems, couplet generation not only considers the sequential context within a sentence line but also emphasizes the relationships between the corresponding words of first and second lines. Therefore, a trapezoidal context character embedding the vector model has been developed firstly, which considers the 'sequence context' and the 'corresponding word context' simultaneously. Afterwards, we chose the typical encoder–decoder framework to solve the sequence–sequence problems, of which the encoder and decoder are used by bi-directional GRU and GRU, respectively. In order to further increase the semantic consistency of the first and second lines of couplets, the pre-trained sentence vector of the first line is added to the attention mechanism in the model. To verify the effectiveness of the method, it is applied to the real data set. Experimental results show that our proposed model can compete with the up-to-date methods, and both adding sentence vectors to attention and using trapezoidal context character vectors can improve the effectiveness of the algorithm.

Keywords: couplet generation model; encoder–decoder framework; gate recurrent unit; word vector

Received 19 February 2020; Revised 18 March 2020; Accepted 28 March 2020
Handling editor: Sabah Mohammed

1. INTRODUCTION

As a unique art form of Chinese language, the couplet is an important part of Chinese traditional culture. It consists of two sentences which are called first line and second line, or upper line and lower line, or antecedent line and subsequent line. They have the same length and are in identical form. Couplet emphasizes the harmony of level and oblique tunes, and neat antithesis, with the intense sense of beauty. The earliest couplets were Spring Festival couplets and then had been widely used in many situations, such as marriage, burial and housewarming.

Generally speaking, given the first line of the couplet, the second line must have the same number of characters with the first one, and the character pairs in the same position should comply with each other [1]. At the same time, the two lines should tell similar or related facts or truths under such strict formal restrictions. Figure 1 shows a couplet, both of the two lines have seven characters, which are 'yard (院) in (内) red (红) plum (梅) play (戏) flying (飞) snow (雪)' and 'door (门) front (前) green (翠) willow (柳) dance (舞) spring (春) breeze (风)', respectively. Taking the first character pair as an example, the two characters are 'yard' and 'door', both of which are parts of a building, while those of the third pair both are color words, and the two characters of the last pair both are weather names. Translated into English, the first line means 'red plum plays snow in the yard', and the second line means 'green willow

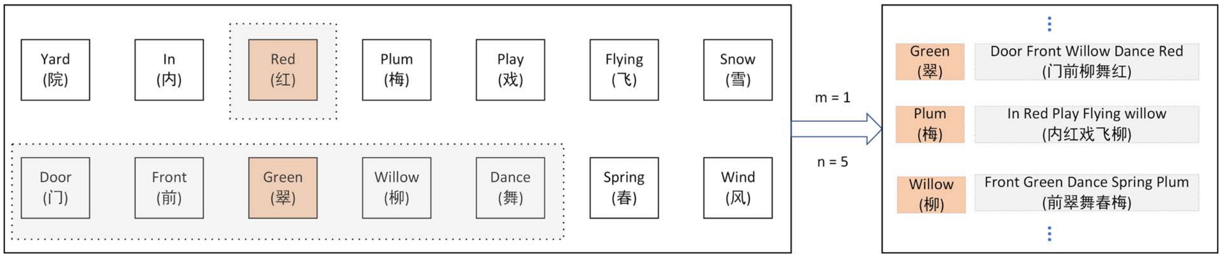

FIGURE 1. Trapezoidal context.

dances with spring breeze in front of the door', depicting beautiful scenes of winter and spring, respectively.

Due to the strict constraints, couplet generation is difficult even for well-educated Chinese people. Therefore, automatic generation of couplets is more challenging, attracting the attention of scholars in the field of natural language processing. Though the research on couplet generation is not extensive yet, there are numerous studies on a very similar topic: automatic generation of poems. The task of couplet generation can be simply regarded as the task of the generation of two lines of a poem, though it is more complicated than the latter because of its strict constraints. The research on automatic generation of couplet or poem could be divided into five classes. (i) The first type of algorithms are rule based methods. For instance, Oliveira *et al*. developed an automatic Portuguese poem generation method and built a platform PoeTryMe based on semantic and grammar templates [2, 3]. References [4, 5, 6] applied rule-based methods to create short classical Japanese poem and haiku. Zhang and Sun classified Chinese couplet rules into hard rules and soft rules and proposed a couplet generation method based on directed probability graph model [7]. (ii) The second type is based on genetic algorithm. For example, Manurung *et al*. defined poetry as a text that satisfies the grammatical rules, semantics and characteristics of poetry and thus regarded the problem of poetry generation as a state-space search problem, and used genetic algorithm to solve it [8]. Yang *et al*. proposed a community competition and adaptive genetic algorithm method for the automatic generation of Tang poems [9]. According to the characteristics of Song Ci, Zhou *et al*. designed the coding algorithm based on the constraints of level and oblique tunes and defined the fitness function by weighting the syntax and semantics and thus constructed a genetic algorithm based model for Song Ci generation [10]. (iii) The third class is based on statistical machine translation. Similar to the translation process, Microsoft Research Asia proposed a statistical machine translation method to generate a lower couplet from the upper and then used the support vector machine to rank the candidate answers [11]. The 'Microsoft couplet' automatic couplet system they constructed has received wide attention. It is notable that they also developed a system 'Microsoft Jueju' for automatic poem writing using similar method. (iv) Next class is based on automatic summarization. Yan *et al*. regarded the task of automatic poetry writing as a constrained optimization problem based on the generative summary framework [12]. (v) The last type of method is based on deep neural networks, especially recurrent neural networks based. Zhang and Lapata jointly performed content selection and surface realization in a recurrent generator framework for Chinese poem generation [13]. Yan formulated the poetry composition task as a natural language generation problem using recurrent neural networks with a polishing schema [14]. Based on the above work, Yan *et al*. proposed an encoder–decoder framework based on recurrent neural networks for Chinese couplet generation [15].

To address the natural language processing problems, the context should be represented by a computer-understood pattern first. Generally, each word is represented by a vector, the simplest mode of which is so called one-hot vector. The one-hot vector is concise and easy to understand. However, it cannot characterize the semantic similarities of different words, which is a critical problem in natural language processing. Thus, many researchers focused on distributed representation trying to use low-dimensional dense vectors to represent words and to depict the correlation between different words. The early distributed representation methods mainly included matrix decomposition methods [16, 17] and probability topic model methods [18, 19]. With the rapid development of deep learning, neural network-based methods have gradually dominated in this field. The first model was developed by Bengio *et al*. in 2003 [20] which was essentially a language model, and as a by-product, the word vector is only a type of trainable parameters, just as the weights of the neural network. Neural network-based methods did not get much attention until Google proposed word2vec in 2013 [21, 22]. After that, Stanford developed GloVe [23], Facebook proposed FastText in 2016 [24] and Peters *et al*. introduced ELMo in 2018 [25].

All of the above methods are based on the assumption that a word can be determined by its context, or vice versa. Therefore, the training set is constructed by sliding a window over the corpus, setting each window as a sample by specifying the central word as the target word and other words as its context. However, unlike normal natural languages, the context of a word or character in couplet, should not only include its context in the same line but also include the words or characters in the other line, especially its counterpart. To address this problem, this paper develops a character embedding vector method based on trapezoidal context. Moreover, besides the character pair's correspondence and the smoothness within a line, we also need to consider semantic consistency between the upper and lower lines. Hence, the sentence information of the first line is added by the attention mechanism in the decoding stage. To test the effectiveness of our proposed method, it is applied to the real datasets. Numerical results show that it can compete with up-to-date methods and both trapezoidal context character embedding vector and the attention of sentence information can improve the performance.

2. COUPLET CHARACTER EMBEDDING VECTOR

Word2vec was created by Google in 2013, which can be conducted efficiently on millions of words and billions of corpus size, and the results can well describe the similarities between words [21]. Nowadays, as one of the most popular tools, word2vec has been widely used in almost all of natural language processing fields. Word2vec has two framework structures, namely CBOW and skip-gram model. It also has two sampling methods, namely hierarchical softmax and negative sampling, respectively. For the detail information of the algorithm, one can refer to [21]. In this paper, CBOW and hierarchical softmax are used to train the embedding vectors.

Traditionally, positive training samples are selected as the pairs of a target word and its context within a given length window, which can be constructed by sliding the given length window over the corpus, for each window specifying the central word as the target word and the others as its context. Therefore, the constructed positive samples satisfy the hypothesis that each word can be represented by its context. While, there is a specificity in Chinese couplet: besides the adjacent words in the same line, each word in couplet has strong relations with the words in the other line, especially the one in the same position. Hence, we proposed the called trapezoidal context character embedding vector method to address this problem. Taking the couplet in Fig. 1 as an example, both lines have seven characters. Then, we define two context windows corresponding to the line of the target word and the other, with the length of n and m, respectively. Assume $n = 5$ and $m = 1$, if the 'Green' is the target word, then the two windows are 'Door Front Green Willow Dance' and 'Red', hence getting the sample as {Green | Door Front Willow Dance Red}. Figure 2 shows the CBOW structure when training the above sample. The input context words are 'Door Front Willow Dance Red', of which each word is represented by a dense vector (the vectors are trainable). Then, they are summarized to the projection layer (only one node), corresponding to the target word 'Green'. For the detailed information of the algorithm, please refer to ref [21].

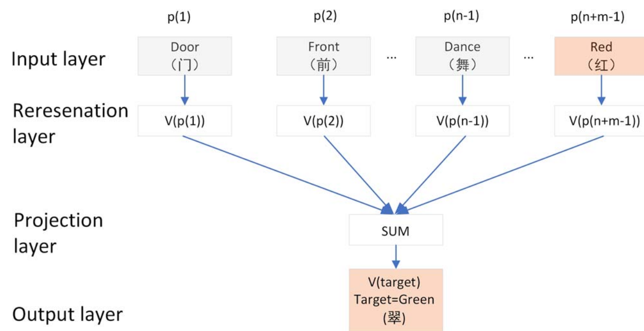

FIGURE 2. Trapezoidal-shape context character embedding the CBOW model.

3. COUPLET GENERATION MODEL

3.1. Framework of the model

With the rapid development of deep learning, the recurrent neural network (RNN) [26, 27] and other improved models—long short-term memory (LSTM) [28, 29] and gate recurrent unit (GRU) [30]—have made a significant contribution to the development of natural language processing. The couplet generation problem can be considered as a sequence-to-sequence problem, which can be well addressed by the commonly used encoder–decoder model [31].

Figure 3 illustrates the network structure of our model. The whole framework adopts encoder–decoder structure, and the coding and decoding models are bi-directional GRU (Bi-GRU) and GRU models, respectively. The coding Bi-GRU model takes as inputs the obtained trapezoidal context character embedding vectors of the first line and produces as output a dense vector for each input character, which is set as input of the attention model together with pre-trained upper line sentence vector. Then, the decoding model will generate the second line according to the given first one.

3.2. Bi-GRU encoder

In our method, the coding model is used as Bi-GRU, including a forward GRU layer and a reverse GRU layer. GRU was developed based on LSTM by integrating the 'forget gate' and 'input gate' as the new 'update gate' and therefore simpler than LSTM [30]. For readers' easy understand, the GRU algorithm will be introduced briefly in the following.

Similar with LSTM, GRU also adopts RNN framework. Figure 4 shows the structure of the unit at time t of GRU. It takes as inputs the input x_t and the previous hidden state $h_{t\text{-}1}$ and produces output the hidden state h_t. In the figure, σ represents the logistic sigmoid function, '+' means concatenation operation. There are two gates within the unit, namely update gate z_t and reset gate r_t. They are calculated by

$$z_t = \sigma\left(\left[W_z x_t\right] + \left[U_z h_{t-1}\right]\right) \quad (1)$$

$$r_t = \sigma\left(\left[W_r x_t\right] + \left[U_r h_{t-1}\right]\right) \quad (2)$$

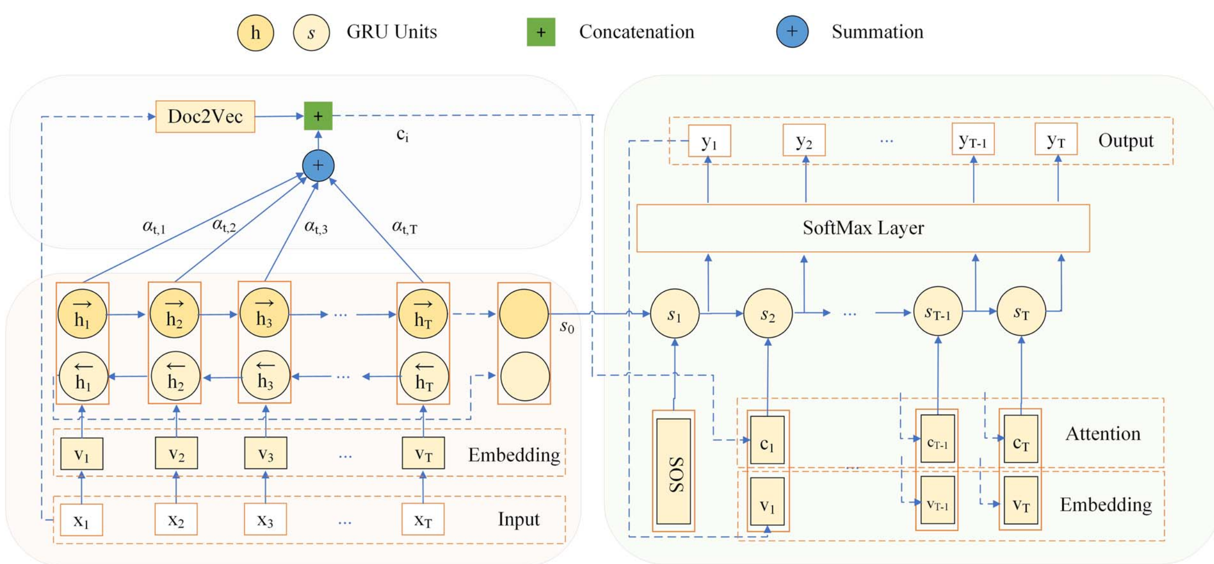

FIGURE 3. The network structure of couplet generation model.

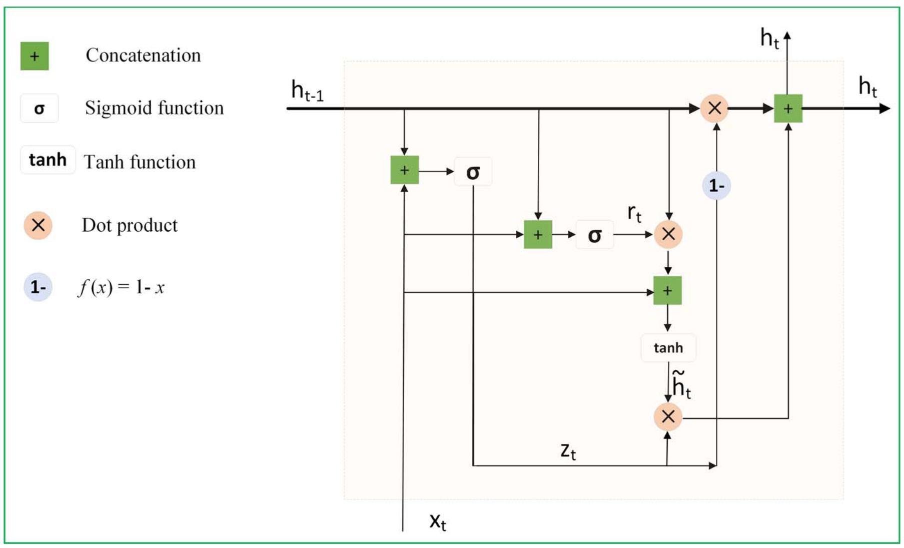

FIGURE 4. The diagram of GRU model.

where W_z, U_z and W_z, U_z are the weights matrix to be learnt.

The actually activation of the proposed unit h_t is then computed by

$$h_t = (1 - z_i)\, h_{t-1} + z_i \tilde{h}_t \tag{3}$$

where

$$\tilde{h}_t = \tanh\left([Wx_t] + \left[U\,(r_t * h_{t-1})\right]\right) \tag{4}$$

similarly, W and U are also the trainable weights matrix.

The Bi-GRU model structure is shown as the lower left in Fig. 3, which includes a forward GRU layer and reverse GRU layer. Then, the output of x_t is

$$\begin{bmatrix} \overrightarrow{h_t} \\ \overleftarrow{h_t} \end{bmatrix}$$

where $\overrightarrow{h_t}$ and $\overrightarrow{h_t}$ are the output hidden state of x_t in forward GRU layer and reverse GRU layer, respectively.

3.3. GRU decoder

The framework of decoder was used as GRU, which is shown on the right of Fig. 3. The GRU unit is the same with shown in Fig. 4. At time t, the unit takes as inputs the concatenation of previous output y_{t-1} and attention result c_t, and the previous GRU unit state s_{t-1}, and produces as output the current state s_t. The current state s_t is taken as input into the next GRU unit, and simultaneously, it is taken through a softmax layer to obtain the corresponding generated character of lower couplet, namely that

$$y_{t,i} = \frac{\exp\left(\theta_i' s_t\right)}{\sum_{j=1}^{N} \exp\left(\theta_j^T s_t\right)} \quad t = 1, 2, \ldots, T; i, j = 1, 2, N \tag{5}$$

where $y_{t,i}$ is the probability of the tth character of second line be the ith character in vocabulary, T the length of the lines, N the vocabulary size, θ_i the embedding vector of the ith character in the vocabulary and s_t output state of the tth GRU unit. Here, s_t can be calculated by Equations (1–4) (the unit state is represented by h in the equations), and the GRU unit structure is the same with shown in Fig. 4. It is notable that the input of each GRU unit is a concatenation vector of the previous obtained character's embedding vector v_{t-1} and the attention result c_t, which will be described in the following subsection. The only exception is the first GRU unit, which takes input as the 'SOS (start of sequence)' token.

3.4. Attention mechanism

To enhance the effect of those inputs having strong relationships with current output, a commonly used method is to use the attention mechanism. For example, Graves proposed a novel attention mechanism [32] that enabled neural networks to focus on different parts of the input sequence, which was improved by Bahdanau *et al.* and successfully applied in machine translation tasks [33]. In the problem of couplet generation, it is easy to understand that the characters in the second line are highly related to the corresponding characters in the same position of the first line. Therefore, we introduced the attention idea in [33] into our model. Numerical results show that it really helps the character alignment between the two lines. However, we found that the semantic affinity of the whole sentence of the two lines was not very satisfactory. To overcome this problem, we introduced the pre-trained doc2vec embedding vectors of the whole first line into the attention mechanism. The attention mechanism structure is shown in Fig. 5.

The final attention result of the tth point s_t is the concatenation of the pre-trained doc2vec embedding vector of first line vd and position specific attention vector $\tilde{c}_t$:

$$c_t = \begin{bmatrix} vd \\ \tilde{c}_t \end{bmatrix} \quad t = 1, 2, \ldots, T \tag{6}$$

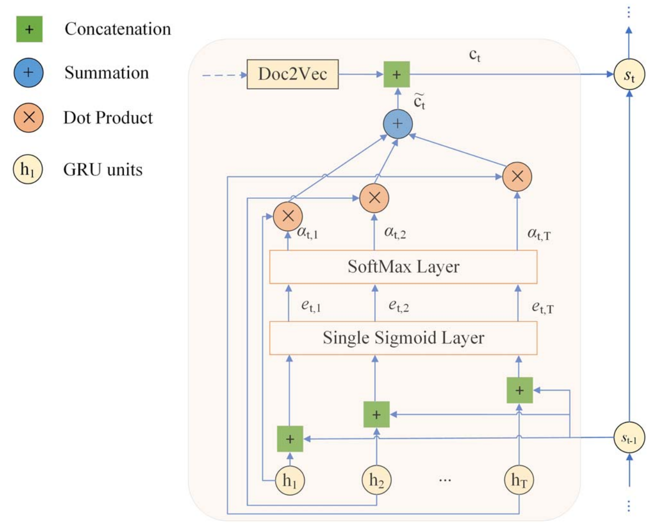

FIGURE 5. The diagram of attention mechanism.

Moreover, the position-specific attention vector $\tilde{c}_t$ is a weighted sum of the states of Bi-GRU on different inputs:

$$\tilde{c}_i = \sum_{j=1}^{T} \alpha_{ij} h_j \tag{7}$$

where the weights α_{ij} is calculated by

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T} \exp\left(e_{ik}\right)} \tag{8}$$

So, weight α_{ij} is normalized e_{ij} by softmax operation. Also, e_{ij} is the output of the concatenation of the (i-1)th state s_{t-1} in decoder and the jth state h_j in encoder through a single sigmoid layer, which can be calculated by

$$e_{ij} = \frac{1}{1 + \exp\left(-W_A \begin{bmatrix} s_{t-1} \\ h_j \end{bmatrix}\right)} \tag{9}$$

where the matrix W_A is trainable attention weights.

4. EXPERIMENT

4.1. Experimental setting

The data sets can be classified into two parts. The first part is collected from different couplet websites, including Chinese couplet library[1], Chinese couplet website[2], Huaxia Couplet

1 http://www.zhgc.com/ylck.asp
2 http://www.duiduilian.com

TABLE 1. The differences of our designed three experiments.

Methods	Representation of inputs	Attention mechanism
Method 1	Embedding vectors trained by word2vec on conventional couplet sentence corpus	No doc2vec in attention (using $\tilde{c}_t$ instead of c_t in Eq. 6)
Method 2	Embedding vectors trained by our proposed trapezoidal context character vectors	Same with Experiment 1
Method 3	Same with Experiment 2	Input doc2vec in attention as the algorithm proposed in the paper

website[3], Zhongqing Shilianwang[4] and Shufaai[5]. However, those collected couplet data size is limited. To enlarge the dataset, we also abstracted strictly aligned pairs in ancient Chinese poetry into our dataset. Finally, the dataset consists of 986 000 pair couplets, which was divided into 18:1:1 for training, verifying and testing, respectively.

In order to verify the effect of our developed model, especially to analyze our proposed trapezoidal context character vectors and the first-line sentence vector in attention, we design three compared experiments. In all these three experiments, we use the encoder–decoder framework, of which the encoder and decoder models are Bi-GRU and GRU, respectively. The differences exist in the representation of inputs and attention mechanism, which are described in Table 1.

'Microsoft couplet' is one of the most popular used automatic couplet system developed by Microsoft Research Asia [11]; therefore, to further test our proposed method, it is also applied to the test data for comparison.

4.2. Evaluation metrics

In recent years, text generation has attracted a lot of attention, but how to evaluate the results is a difficult problem. Generally speaking, human evaluation is the most reasonable method, but it requires much time and labor costs. Therefore, after human evaluation on small test data, we use other four automatic evaluation metrics for comparison, namely that BLEU, Meteor, Rouge and CIDEr.

BLEU (Bilingual Evaluation Understudy) [34] evaluates candidate sentences based on n-tuple matching between the candidate sentences and their ground truths, where n is usually taken with a value from 1, 2, 3 or 4. BLEU is considered as one of the closest evaluation metrics to human evaluation. In our experiments, n is taken with two values, namely 1 and 2.

ROUGE [35] is similar to BLEU and is also based on n-tuple matching, except that it takes into account recall. ROUGE includes ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. ROUGE-L is used in this paper, which is based on the recall of the longest common string (LCS) co-occurrence and F-measure.

METEOR [36] is based on an explicit word-to-word matching between the candidate output and its ground truth. It supports not only matching between words that are identical in the two strings being compared, but can also match words that are simple morphological variants of each other.

CIDEr [37] is originally proposed to solve the problem of picture abstract. CIDEr takes into account the frequency of n-tuples in the corpus and considers that those with high frequencies have less information and are less important. It treats each sentence as a 'document', expressing as a **tf-idf** vector, and then calculates the cosine similarity between the sentence to be evaluated and its reference.

For human evaluation, we invited 10 well-educated on ancient Chinese poetry and couplet graduated students to evaluate the generated lower lines of experiment1, experiment2, experiment3 and 'Microsoft couplet' [11] on random selected 100 test data. Scores are required between 0 and 10, the bigger the better.

4.3. Experimental results

Firstly, our proposed methods are compared with 'Microsoft couplet' [11] on random selected 100 couplets according to human evaluation. Table 2 shows the generated results of 10 couplets, from which we can roughly compare with the performance of different methods. In the table, the same colored characters of upper line and corresponding lower lines produced by different methods means that they are very well matched both in format and in semantic. In the first couplet, the result of method 1 has no antithesis to the upper line, for example 'slim(纤)' and 'finger(指)' totally cannot match with 'green(翠)' and 'gold(金)' of the upper line, the other three are much better, especially 'Microsoft couplet' and method 3. In terms of semantic correlation between the upper and lower lines, the results of method 2 and method 3 are significantly stronger than the other two methods. The results of the third couplet are very similar with the first one: the antithesis of the method 1 is the worst one; in terms of semantic correlation, the other three methods are also better than method 1, while among

3 http://www.aiduilian.cn
4 http://bbs.zqslw.com
5 http://www.shufaai.com

TABLE 2. Part of results. (For interpretation of the references to colour, the reader is referred to the web version of this article.)

Upper line	Result of Microsoft couplet[11]	Result of Method 1	Result of Method 2	Result of Method 3
The emerald gold pressed against his face 翠钿金压脸	Green chardonnay king is happy 青霞王开心	Slender hands against slender waist 纤手指纤腰	The pink jade is her eyebrows 红粉玉成眉	The red jade is attached to the lips 红绣玉衔唇
The bright moon rose from the tian mountains 明月出天山	General sima sang 司马将军歌	A thousand miles of pale earth 苍苍千万里	In the pale smoke 苍苍烟雾里	In the vast white clouds 苍茫白云里
The city beneath the white clouds and the mountains 白云峰下城	The moon before the green water and hills 绿水山前月	The bright moon and pale moss 明月苍苍苔	Grass and alleys in the city 青草城中堂	Sky in green water and caves 碧水洞中天
A grain in the pot becomes something wonderful. 壶中一粒化奇物	A few dots in front of eyes become beauties 眼前几点成佳人	Everything is blue peony 万物皆皆青牡丹	The world is full of profusion 世间纷纷满天地	Thounds of words under the pen contains mysterious 笔底千般蕴妙机
Where to sleep because of cold days and wide fields? 天寒野旷何处宿	How much time in clam water and world? 水静红尘几多时	Leaf of *Malus spectabilis* whizz 棠梨花叶飕飕飕	Leaf of malus spectabilis whizz 棠梨叶风飕飕飕	Here is quiet on vast land and quiet mountain 地阔山静此处幽
Half of musk mountains 麝香山一半	The moon in the third watch is like milk 牛奶月三更	It's not like flowing through water 不似水中流	The clear water and green clouds 云水碧云清	Third cloud cold moon 云冷月三分
Want to find the top of mount lushan 欲寻庐峰顶	In order to get the heart of the cloud 为有径云心	Do not wait for the mind of the white cloud 不待白云期	Hope to get the heart of the white clouds 乞与白云期	Leave for Luoyang city 更往洛阳城
People are infatuated with dongting-water 人迷洞庭水	Geese flying over xiaoxiang smoke 雁度潇湘烟	Birds fly over the water in the dong-ting 鸟入洞庭波	Tears fell on the xiangriv sky 泪滴湘江天	Guests are intoxicated in whitecloud-mountain 客醉白云山
What never changes is the coldness of winter 岁寒终不变	Under the moon he has become self-contained 月下已自成	Look forward to the coquettish in morning and evening 朝夕待风骚	The grass has withered, who knows 芳草竟谁知	What is impermanent is the warmth of spring 春暖总无常
The pine tree falls, the crane dies, the mulberry field changes 松倾鹤死桑田变	The water fell and the flowers disappeared and the wine came up 水落花无酒上来	White hair grows year after year like wicker 柳带年年白发生	The cow and the sheep sigh at the distance 石上牛羊泣路长	The rain fell and the flowers appeared and grass grow 雨落花生草木长

of which, the results of 'Microsoft couplet' and method 3 are much smoother. For the eighth couplet, two characters in the result of method 1 are repeated with the.

upper couplet (Not meet the couplet rules), and the result of method 2 is obviously less semantic fluency than the other methods. To be noted that the result of 'Microsoft couplet' is exactly the original lower line. This is because 'Microsoft couplet' will first retrieve the query upper line in its couplet library, which happens to include the eighth couplet pair. However, the result of Method 3 is also very good in terms of both antithesis and semantics, and all the evaluators considered it is as good as the original one, or even better. For the last couplet, the results of 'Microsoft couplet' and Method 3 are significantly better than the other two methods, and Method 3 is more tunable. Therefore, from Table 2, it could be concluded that Method 1 is the worst one on performance, 'Microsoft couplet' and Method 3 are better than the other two methods, of which Method 3 is a little better.

TABLE 3. Artificial evaluation of different methods.

Evaluator	Microsoft couplet [11]	Method 1	Method 2	Method 3
1	4.64	4.65	4.66	4.66
2	4.63	4.61	4.62	4.63
3	4.68	4.69	4.70	4.71
4	4.80	4.76	4.76	4.76
5	4.64	4.56	4.57	4.68
6	4.66	4.66	4.68	4.68
7	4.92	4.80	4.80	4.81
8	4.81	4.70	4.78	4.80
9	4.80	4.81	4.81	4.83
10	4.82	4.88	4.89	4.90
Average	4.74	4.66	4.72	**4.75**

The italicized and bold number is the best of the three comparison method results.

TABLE 4. Comparison results of our proposed different frameworks.

Evaluation method	Method 1	Method 2	Method 3
BLEU1 (%)	16.32	*16.66*	**16.92**
BLEU2 (%)	13.64	*13.90*	**14.05**
Meteor (%)	12.79	*12.96*	**13.06**
Rouge_L (%)	16.44	*16.67*	**16.89**
CIDEr (%)	35.58	*37.41*	**37.57**

The italicized and bold number is the best of the three comparison method results.

Table 3 shows the manual-evaluated comparison of the results of four comparison methods of randomly selected 100 test data. The average score of 'Microsoft couplet' and Methods 1, 2 and 3 are 4.74, 4.66, 4.72 and 4.75, respectively. Therefore, we can conclude that our proposed model (Method 3) is as good as 'Microsoft couplet', or even better, and both 'Trapezoidal Context' character vector and sentence vector attention mechanism are effective.

Table 4 shows the comparison results of our proposed different frameworks on metrics of BLEU1, BLEU2, Meteor, Rouge_L and CIDEr, of which BLEU1 and BLEU2 mean that the matching n-tuples in BLEU are set as 1-tuple and 2-tuple, respectively. Since the results of 'Microsoft couplet' only can be obtained online and large-scale automatic evaluation cannot be conducted, they are not included in Table 4. As can be seen from the table, the results of Method 3 are the best in terms of all metrics, and the results of Method 2 are the second. This indicates that both 'Trapezoidal Context' character vector proposed in this paper and adding the first-line sentence vector in attention mechanism can improve the quality of the generated lower couplet.

5. CONCLUSION

The couplet is an excellent symbol of Chinese traditional culture, and the automatic generation of couplets is a useful complement of it. This paper proposes an encoder–decoder framework for couplet generation. In the model, the encoder and decoder are used as Bi-GRU and GRU, respectively. Compared with the existing methods, there are two main aspects of innovation. (i) Considering the particularity of couplet, a character vector model based on 'Trapezoidal Context' is developed to contain the semantic relation between the corresponding position characters of upper and lower lines. (ii) To enhance the semantic correlation of generated lower lines with upper lines, the sentence vector of the upper line is added into the attention mechanism. The experimental results show that our proposed model is as good as the up-to-date existed method, or even better, and both 'Trapezoidal Context' character vector and adding upper line sentence vector into attention improve the effectiveness of our model.

Acknowledgements

The authors are grateful to the support of the National Natural Science Foundation of China (61972174).

REFERENCES

[1] Jiang, L. and Zhou, M. (2008) Generating Chinese Couplets using a Statistical MT Approach. International Conference on Computational Linguistics. In *Proceedings of COLING 2008,*

Manchester, UK, 18–22 August, pp. 377–384. Association for Computational Linguistics, Stroudsburg.

[2] Oliveira, H.G., Hervas, R., Diaz, A. and Gervas, P. (2017) Multilingual extension and evaluation of a poetry generator. *Nat. Lang. Eng.*, 23, 929–967.

[3] Oliveira, H.G. (2017) A Survey on Intelligent Poetry Generation: Languages, Features, Techniques, Reutilisation and Evaluation. In *Proceedings of 10th International Conference on Natural Language Generation. Santiago de Compostela, Spain, 4–7 September*, pp. 11–20. Association for Computational Linguistics, Stroudsburg.

[4] Tosa, N., Obara, H. and Minoh, M. (2008) Hitch haiku: an interactive supporting system for composing haiku poem. *International Federation for Information Processing*, 5309, 209–216.

[5] Wu, X., Tosa, N. and Nakatsu, R. (2009) New Hitch Haiku: An Interactive Renku Poem Composition Supporting Tool Applied for Sightseeing Navigation System. In *Proceedings of ICEC 2009, Paris, France, 3–5 September*, pp. 191–196. Springer-Verlag, Berlin.

[6] Netzer, Y., Gabay, D., Goldberg, Y. and Elhadad, M. (2010) Gaiku: Generating Haiku with Word Associations Norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity 2009, Boulder, Colorado, 4 June*, pp. 32–39. Association for Computational Linguistics, Stroudsburg.

[7] Zhang, K.X. and Sun, M.S. (2009) A Chinese couplet generation model based on statistics and rules. *J. Chinese Inf. Process.*, 23, 100–109.

[8] Manurung, R., Ritchie, G. and Thompson, H. (2012) Using genetic algorithms to create meaningful poetic text. *J. Exp. Theor. Artif. Intel.*, 24, 43–64.

[9] Yang, W.J., Cheng, Y.N., He, J., Hu, W.Q. and Lin, X.J. (2016) Research on community competition and adaptive genetic algorithm for automatic generation of tang poetry. *Math. Probl. Eng.*, 2016, 4076154.

[10] Zhou, C.L., Wei, Y. and Ding, X.J. (2010) Genetic algorithm and its implementation of automatic generation of Chinese SONGCI. *J. Softw.*, 21, 27–437.

[11] He, J., Zhou, M. and Jiang, L. (2012) Generating Chinese classical poems with statistical machine translation models. In *Proceedings of Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Canada, 22–26 July*, pp. 1650–1656. AAAI Press, Menlo Park, California.

[12] Yan, R., Jiang, H., Lapata, M., Lin, S.D., Lv, X.Q. and Li, X.M. (2016) i, poet: automatic Chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August*, pp. 2197–2203. AAAI Press, Menlo Park, California.

[13] Zhang, X. and Lapata, M. (2014) Chinese poetry generation with recurrent neural networks. In *Proceedings of EMNLP 2014, Doha, Qatar, 25–29 October*, pp. 670–680. Association for Computational Linguistics, Stroudsburg.

[14] Yan, R. (2016) i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August*, pp. 2238–2244. AAAI Press, Menlo Park, California.

[15] Yan, R., Li, C.T., Hu, X.H. and Xhang, M. (2016) Chinese Couplet Generation with Neural Network Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August*, pp. 2347–2357. Association for Computational Linguistics, Stroudsburg.

[16] Dumais, S. (1990) Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41, 391–407.

[17] Lee, D.D. and Seung, H.S. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788–791.

[18] Hofmann, T. (1999) Probabilistic latent semantic indexing. In *Proceedings of SIGIR 99, Berkeley California USA, 15–19 August*, pp. 50–57. Association for Computing Machinery, New York.

[19] Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) Latent dirichl*et al*location. *J. Mach. Learn. Res.*, 3, 993–1022.

[20] Bingio, Y., Ducharm, R., Vincent, P. and Jauvin, C. (2003) A neural probabilistic language model. *J. Mach. Learn. Res.*, 3, 1137–1155.

[21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013, Lake Tahoe, Nevada, 5–10 December*, pp. 3111–3119. MIT Press, Cambridge, MA.

[22] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient estimation of word representations in vector space. In *Proceedings of ICLR 2013, Scottsdale, Arizona, 2-4 May*. https://\ignorespacesarXiv:1301.3781.

[23] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. In *Proceedings of the EMNLP 2014, Doha, Qatar, 25–29 October*, pp. 1532–1543. Association for Computational Linguistics, Stroudsburg.

[24] Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T. (2017) Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April*, pp. 427–431. Association for Computational Linguistics, Stroudsburg.

[25] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (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, New Orleans, Louisiana, 1–6 June*, pp. 2227–2237. Association for Computational Linguistics, Stroudsburg.

[26] Mikolov, T., Karafiát, M., Burget, L., Cernocky, J. and Khudanpur, S. (2010) Recurrent Neural Network Based Language Model. Proceedings of the Conference of the International Speech Communication Association 2010. In *Makuhari, Chiba, Japan, 26–30 September*, pp. 1045–1048. International Speech Communication Association, Baixas, France.

[27] Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural Comput.*, 9, 1735–1780.

[28] Cho, K., Van Merrienboer, B., Bahdanau, D. and Bengio, Y. (2014) On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Proceedings of SSST-8. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation 2014, Doha, Qatar, 25 October*, pp. 103–111.

Association for Computational Linguistics, Stroudsburg.

[29] Sundermeyer, M., Schlüter, R. and Ney, H. (2012) LSTM Neural Networks for Language Modeling. In *Proceedings of Interspeech 2012, Portland, OR, 9–13 September*, pp. 601–608. Association for Computational Linguistics, Stroudsburg.

[30] Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the EMNLP 2014, Doha, Qatar, 25–29 October*, pp. 1724–1734. Association for Computational Linguistics, Stroudsburg.

[31] Sutskever, I., Vinyals, O. and Le, Q.V. (2014) Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS 2014, Montreal, Quebec, Canada, 8–13 December*, pp. 3104–3112. MIT Press, Cambridge, MA.

[32] Graves A. (2018) Generating Sequences with Recurrent Neural Networks. [2018-07-18] https://arxiv.org/abs/1308.0850

[33] Bahdanau, D., Cho, K. and Bengio, Y. (2015) Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR 2015*, pp. 7–9, San Diego, CA, USA, 7-8 May. pp. 7-9. https://\ignorespacesarXiv:1409.0473.

[34] Papineni, K., Roukos, S., Ward, T. and Zhu, W.J. (2002) BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 19–24 July 2002*, pp. 311–318. Association for Computational Linguistics, Stroudsburg.

[35] Lin, C.Y. (2004) ROUGE: A Package for Automatic Evaluation of Summaries. Proceedings of Workshop on Text Summarization Branches Out. In *Post-Conference Workshop of ACL 2004, Barcelona, Spain, July*, pp. 74–81. Association for Computational Linguistics, Stroudsburg.

[36] Denkowski, M. and Lavie, A. (2014) Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation 2014, Baltimore, Maryland, 26–27 June*, pp. 376–380. Association for Computational Linguistics, Stroudsburg.

[37] Vedantam, P., Zitnick, C.L. and Parikh, D. (2015) CIDEr: Consensus-based Image Description Evaluation. In *Proceedings of the CVPR 2015, Boston, MA, 7–12 June*, pp. 4566–4575. IEEE Computer Society, Washington, DC.