IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON RECENT COMPUTATIONAL METHODS IN KNOWLEDGE ENGINEERING AND INTELLIGENCE COMPUTATION

Received October 10, 2017, accepted November 6, 2017, date of publication November 17, 2017, date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2774839

Automatic Generation of News Comments Based on Gated Attention Neural Networks

HAI-TAO ZHENG[1], WEI WANG[1], WANG CHEN[1], AND ARUN KUMAR SANGAIAH[2]
[1]Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[2]School of Computing Science and Engineering, VIT University, Vellore 632014, India

Corresponding author: Wei Wang (w-w16@mails.tsinghua.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61773229, in part by the Natural Science Foundation of Guangdong Province under Grant 2014A030313745, in part by the Basic Scientific Research Program of Shenzhen City under Grant JCYJ20160331184440545, and in part by the Cross fund of Graduate School at Shenzhen, Tsinghua University under Grant JC20140001.

ABSTRACT With the development of recurrent neural networks (RNN), various natural language generation (NLG) tasks have boomed in the past few years, such as response generation in conversation and poetry generation. However, automatic generation of news comments is a new, challenging and not well-studied task in NLG. Different from other NLG tasks, this task requires the contextual relevance between comments and news. In addition, we need to generate diversified comments, because different people usually have different opinions on the same news in the real world. In this paper, we propose a gated attention neural network model (GANN) to generate news comments. To address the problem of contextual relevance, we introduce the gated attention mechanism to use news context self-adaptively and selectively. To ensure the diversity of comments, we use random sample and relevance control to generate comments with different topics and degrees of relevance. Moreover, we apply generative adversarial nets to improve GANN. Automatic evaluation with perplexity score reveals that GANN outperforms the existing comment generation methods. Human evaluation proves that the generated news comments are close to human comments.

INDEX TERMS Recurrent neural networks, natural language generation, natural language processing, artificial intelligence.

I. INTRODUCTION

Natural Language Generation (NLG) belongs to the subtopic of artificial intelligence and computational linguistics. The aim of NLG is generating understandable texts in human languages [1]. The progress achieved in NLG will contribute to building strong intelligent system that can comprehend and compose human languages. Traditional approaches of NLG have been applied successfully in some fields, such as automatic generation of news [2], weather reports [3] and questions [4]. Recently Recurrent Neural Networks (RNN) has shown promising performance in textual generation [5]–[7] and language modeling [8]–[10]. Compared to traditional approaches, the RNN based approaches learn the generating function from text data automatically. This advantage makes an increasing number of researchers explore a variety of NLG tasks, such as poetry generation [11]–[13] and response generation in conversation [14], [15].

In this study, we focus on the task of automatic generation of news comments for news articles, which is a new, challenging and not well-studied task in NLG. Fig. 1 shows some examples of real comments. This task is helpful to comprehend human languages, like how people write news comments, what people pay attention to, and how people express it. Moreover, it is also a useful exploration for the future task of automatic generation of comment article. Additionally, these generated comments are beneficial for companies. For example, we could build a comment writing assistant which generates some candidate comments for users. Users could select one and refine it, which makes the procedure more user-friendly.

Several previous studies have attempted for comment generation. Tang *et al.* [16] proposed a context-aware model to generate product comments from product number and rate. Similar to Tang *et al.* [16], Costa *et al.* [17] studied the task of generating product explanations according to a set of rates. Dong *et al.* [18] proposed an attention based model to generate product comments. However, these work focus on generation of product comments. And they use relatively simple context, such as rate score, whereas we focus on generation of news comments, which has more complex context.

FIGURE 1. Examples of real comments.

There are two challenges in comment generation. First, it should ensure the contextual relevance between comments and news. This involves complex contextual information to deal, such as a news event which generally contains many aspects, e.g., time, location and people. The generated comments should be relevant to at least one aspect of news in semantics. Second, it requires generating diversified comments for each news, which makes the task more challenging. Generally, different users tend to comment on different aspects of the same news article.

In this paper, we propose a Gated Attention Neural Network model (GANN) to generate news comments. GANN consists of two parts: comment generator and comment discriminator. The comment generator is built on an encoder-decoder framework and we introduce the gated attention mechanism and relevance control to boost it. Inspired by the idea of generative adversarial nets, we apply an extra discriminator which distinguishes real and fake comments to improve the comment generator.

To address the problem of contextual relevance, we introduce the gated attention mechanism in the decoding phase. The gated attention mechanism learns soft alignments between generated words and news context, and adaptively computes encoder-side context vectors. Moreover, the gate dynamically selects the amount of contextual information to predict the next word. To address the problem of diversity, we use random sample and relevance control. The random sample guarantees that comments have diverse topics. And the relevance control contributes to generating comments with different degrees of relevance.

In order to evaluate our method, we crawl a news comment corpus from web and perform experiments on it. Automatic evaluation with perplexity score reveals that GANN outperforms the existing comment generation methods. Furthermore, human evaluation proves that the generated news comments are close to human comments.

The main contributions of this paper are summarized as follows:

- We first investigate the task of generation of news comments and GANN introduces the gated attention mechanism to use news context self-adaptively and selectively, which ensures the contextual relevance.
- We generate diversified comments with different topics and different degrees of relevance by utilizing random sample and the relevance control.
- We use generative adversarial nets to improve the comment generator in order to generate more natural comments.
- Experimental results on real comment dataset show that GANN achieves better performance than the existing comment generation methods.

The remainder of this paper is organized as follows. We review related works in Section 2. Section 3 introduces the detail of our proposed model. Section 4 describes dataset, experimental setups and gives the result analysis. Finally, we give a summary of this paper and discuss the future direction of improvement in Section 5.

II. RELATED WORK

Comment generation is a subtask of NLG. Conventional approaches of NLG typically divide the task into sentence planning and surface realization [1]. Sentence planning maps inputs into an intermediary form representing the utterance, then surface realization converts the intermediate structure into the final text [19], [20]. Although these rule-based approaches have been well explored previously, they still need to define many rules [21] and only apply some specific tasks and domains. Moreover, the languages generated by these approaches are rigid and lack of the large variations of human languages. Although some approaches are proposed to learn the template from corpus automatically [22], it is very expensive to get the data to train the model and they still need many human handcrafted features.

The RNN based approaches of NLG have drawn more and more attention in recent years. Compared to the traditional rule-based approaches, the RNN based approaches overcome above shortcomings and provide an end-to-end solution without much human participation. Sutskever *et al.* [6] proposed multiplicative RNN model to predict the next word, which uses different transformation functions for different characters. Bowman *et al.* [7] generated sentences from continuous semantic spaces with a variational auto-encoder. Mikolov and Zweig [9] and Wang and Cho [10] improved the performance of language modeling through the long dependency of RNN. Their methods outperformed the n-gram language modeling significantly. These work prove the effectiveness of RNN in text generation and our work follows this research line.

Many researches of other NLG tasks are also related to our work. Zhang and Lapata [11], Wang *et al.* [12] and

Wang *et al*. [13] used RNN to solve the poetry generation task, in which the poetry lines are generated sequentially by accumulating the status of the lines that have been generated so far. However, these work have a simple context, for instance a keyword input by user. Wen *et al*. [15] used RNN to generate responses in conversation. Compared to us, they use explicit slot values as context.

Several previous studies have attempted for comment generation. Tang *et al*. [16] proposed a context-aware model to generate product comments from product number and rate, which are used as contextual information. Similar to Tang *et al*. [16], Costa *et al*. [17] studied the task of generating product explanations according to a set of rates. Dong *et al*. [18] proposed a RNN and attention based model which generates comments depending on product number, rate and user number. However, these work focus on generation of product comments and use relatively simple context, such as rate score. Furthermore, the words used by users are very similar and the topics of comments are much less, usually about product quality and appearance. Compared to these work, we focus on generation of news comments, in which we have relatively complex contextual information. And the topics of comments are diversified.

III. METHOD

To begin with, we state the problem of generation of news comments as follows: given the news title $x = x_1, x_2, \ldots, x_m$ as input, the model needs to generate relevant news comment $y = y_1, y_2, \ldots, y_n$ maximizing the conditional probability $p(y|x)$. The length of the entire news is commonly long, and there is a lot of content-independent redundant information. We need to select the important information of the news. The title of news highly summarizes the entire content of news, thus we use it as input.

As shown in Fig. 2, our proposed GANN model contains two components: comment generator and comment discriminator. The generator uses an encoder-decoder framework (see Fig. 3). Inspired by the idea of generative adversarial nets, we add an extra discriminator to improve the comment generator. In order to ensure the contextual relevance, we introduce the gated attention mechanism in the decoding phase (see Fig. 4). The gated attention mechanism learns soft alignments between generated words and news context, and adaptively computes encoder-side context vectors. Moreover, the gate dynamically selects the amount of contextual information to predict the next word. In order to generate diversified comments, we use random sample strategy to generate comments with diverse topics. And we utilize the relevance control to generate comments with different degrees of relevance. We first introduce the background of RNN, and then describe our model.

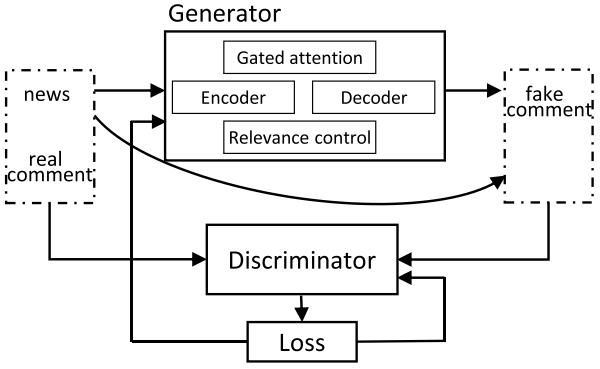


FIGURE 2. Gated Attention Neural Network model.

A. BACKGROUND: RECURRENT NEURAL NETWORKS

Recurrent neural networks (RNN) processes sequence information well, which represents history information into a hidden state and then computes a probability distribution of next word according to the hidden state. The computing process is as follows:

$$h_t = \tanh\left(Wh_{t-1} + Vx_{t-1}\right), \tag{1}$$

$$p(x_{t+1}|h_t) \propto \exp\left(Oh_t\right), \tag{2}$$

where the hidden state h_t summaries the history information, W, V and O are weight params, $p(x_{t+1}|h_t)$ represents the probability of next word.

The overall probability of a sequence $x = x_1, x_2, \ldots, x_T$ is calculated as follows:

$$p(x) = \prod_{t=1}^{T} p(x_t|h_{t-1}). \tag{3}$$

Training RNN can be done through maximizing the joint probability $p(x)$ defined by Equation 3. However, training the basic RNN above suffers from the problem of gradient vanishing or exploding. The long-short term memory (LSTM) unit [23] addresses the problem effectively. The core idea of LSTM is introducing the memory state and multiple gating functions to control the information written to the memory sate, read from the memory state, and removed from the memory state. The computing process of LSTM is as follows:

$$\begin{aligned} i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1}\right) \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1}\right) \\ o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1}\right) \\ g_t &= \tanh\left(W_{xc}x_t + W_{hc}h_{t-1}\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh\left(c_t\right), \end{aligned} \tag{4}$$

where σ is the sigmoid function, i_t, f_t, o_t are input, forget, and output gates respectively, and g_t, c_t are proposed cell value and true cell value, h_t is the new hidden state.

B. COMMENT GENERATOR

1) NEWS ENCODER

The model converts all words $x_1, x_2, \ldots, x_m$ of title into one-hot vectors and obtains the embedding representations by

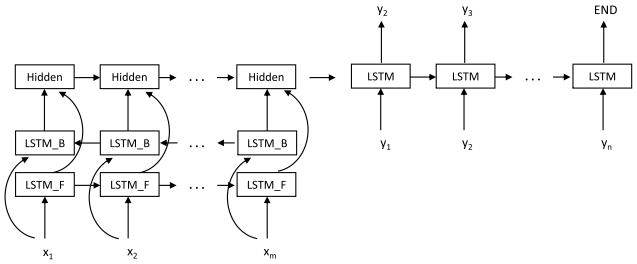


FIGURE 3. News encoder and comment decoder.

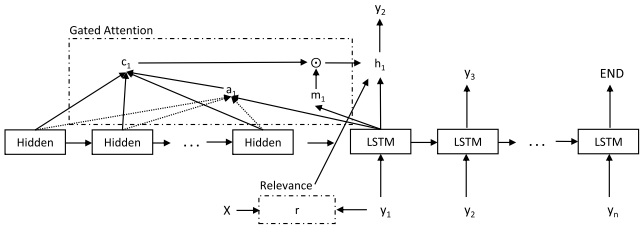


FIGURE 4. Gated attention and relevance control.

multiplying the embedding matrix. The embedding representation of x_i is computed as follows:

$$\vec{e}_i = E\vec{x}_i, \tag{5}$$

where $E \in \mathbb{R}^{l\times|V|}$ is the embedding matrix, l is the dimension of embedding, and $|V|$ is the vocabulary size. Then these embedding vectors are fed into encoder one by one to compute the forward hidden vectors via:

$$\overrightarrow{h_i} = LSTM_f(\vec{e}_i), \tag{6}$$

where $\overrightarrow{h_i} \in \mathbb{R}^k$ is a k-dimension hidden vector and $LSTM_f$ denotes the LSTM unit. At the same time, the reversed sequence is fed into backward LSTM to get backward hidden vectors:

$$\overleftarrow{h_i} = LSTM_b(\vec{e}_i), \tag{7}$$

where $\overleftarrow{h_i} \in \mathbb{R}^k$ is a k-dimension hidden vector. Concatenating them we get the final hidden vectors of all words:

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}], \tag{8}$$

where $h_i \in \mathbb{R}^{2k}$ is the final hidden vector of word x_i. The hidden vector of last word is used as the representation of the title $h_c = h_m$.

2) COMMENT DECODER

The last hidden vector of the title is used to initialize the decoder. Similar to encoder, the model coverts the sequence of comment words $y_1, y_2, \ldots, y_n$ into one-hot vectors and gets their low-dimensional representations through the shared embedding matrix E:

$$\vec{e}_j = E\vec{y}_j. \tag{9}$$

Notice that we use the identical embedding matrix when encoding and decoding. Then, feeding these embedding vectors to decoder to compute hidden vectors via:

$$s_j = LSTM(\vec{e}_j), \tag{10}$$

where $s_j \in \mathbb{R}^k$ is the hidden vector of comment word. Now according to the vanilla decoder without using an attention mechanism, the model computes the probability of the next word via:

$$p(y_{j+1}|y \le j) = p(y_{j+1}|s_j) \propto \exp\left(Ws_j\right), \tag{11}$$

where $W \in \mathbb{R}^{|V|\times k}$ is a parameter matrix. Based on the next word and the actual word we have, we can calculate the cross-entropy of the generated comment sequence for model training.

3) GATED ATTENTION

In order to guarantee the contextual relevance between comments and news, we need to handle the contextual information efficiently. The different words in the comment concentrate on different parts of the title context, hence the model should use the context self-adaptively. However, as indicated in Equation 11 the vanilla decoder ignores this problem. To better utilize contextual information we introduce the attention mechanism, which assigns different weights to different parts of contextual information. Previous work has proved that it significantly improves performance [24].

Moreover, not all comment words require using the context when decoding. Some words are merely decided by preceding words. Therefore, using the title context for all words may be problematic. We tackle this problem though the gated attention mechanism which introduces a selective gate to control the amount of contextual information for reference in original attention mechanism.

First, we compute the attention scores between the current hidden vector and all hidden vectors of title words and normalize them to get the final attention weights.

$$d_i = v^T \tanh\left(W_h h_i + W_s s_j\right), \tag{12}$$

$$a_i = \frac{\exp\left(d_i\right)}{\sum_{k=1}^{m} \exp\left(d_k\right)}, \tag{13}$$

where $v \in \mathbb{R}^{2k}$ is a parameter vector, $W_h \in \mathbb{R}^{2k\times 2k}$ and $W_s \in \mathbb{R}^{2k\times k}$ are parameter matrices. Next, the new context vector $\tilde{h}_c$ is obtained by:

$$\tilde{h}_c = \sum a_i h_i, \tag{14}$$

where $\tilde{h}_c \in \mathbb{R}^{2k}$. Then, we employ a gating function which products a score range 0-1 as a weight depending on the current decoder hidden vector via:

$$m_j = \sigma\left(Gs_j + b\right), \tag{15}$$

where $G \in \mathbb{R}^{2k\times k}$, $\sigma(\cdot)$ is the sigmoid function. The new hidden vector is obtained by:

$$\tilde{s}_j = W_g[s_j, m \odot \tilde{h}_c] + b, \tag{16}$$

where $W_g \in \mathbb{R}^{k\times 3k}$, the brackets [,] denote concatenation. By using the new hidden vector the model can emphasize different parts of the title when generating each word. It can be seen that in the decoding phase, not only the contextual information is adaptively used, but also the information is selectively used.

4) RELEVANCE CONTROL

Different from other NLG tasks which have one-to-one relation, there are diversified comments for each news, especially the popular ones. And the relevance between these comments and news are far from each other. For example, there are some general comments, such as ''I don't know.'' These comments have little relevance to the news and could correspond to various news. In addition, there are some detailed comments which are closely related to the news. However, the vanilla decoder does not model this. In order to generate diversified comments in relevance, we use a relevance vector to control the degree of relevance when decoding.

We need to assign a relevance level to each comment according to its relevance to news. We use a simple method, in which more words overlap means more relevant. Through this method, we get a one-hot relevance vector r of three-level degrees according to the number of common words. When getting the decoder hidden vector $\tilde{s}_j$, we operate on it with relevance vector to get a new hidden vector via:

$$s_r = \tanh\left(W_1 r + W_2 \tilde{s}_j\right), \tag{17}$$

where $W_1 \in \mathbb{R}^{k\times 3}$, $W_2 \in \mathbb{R}^{k\times k}$. Then s_r contains not only the history information but also the degree information of relevance, we can use it to predict the next word via:

$$p(y_{j+1}|y \le j) \propto \exp\left(W s_r\right). \tag{18}$$

C. COMMENT DISCRIMINATOR

Generative adversarial nets(GAN) proposed by Goodfellow *et al*. [25] is a promising framework for many tasks. This approach has been successful and been mostly applied in computer vision tasks of generating samples of natural images [26]. Inspired by the idea of adversarial, we add a comment discriminator to improve the generator. However, GAN is designed for generating real valued, continuous data but has difficulty in directly generating sequences of discrete tokens. Yu *et al*. [27] use policy gradient reinforcement learning to backpropagate the error from the discriminator which naturally avoids the differentiation difficulty for discrete data. In this paper, we choose the CNN as our discriminator because CNN has recently been shown of great effectiveness in text classification [28]. The discriminator adopts a structure similar to Yu *et al*. [27] and we also use policy gradient reinforcement learning for adversarial training.

We first get embedding presentation of title words and comments words. Then we concatenate all of them into one vector.

$$X = [e_1, e_2, \ldots, e_m, e_{y1}, \ldots, e_{yn}], \tag{19}$$

where $e_i \in \mathbb{R}^l$ is the embedding of word and [., .] is the concatenation operator. After that, a kernel $W \in \mathbb{R}^{k\times l}$ applies a convolutional operation to produce a new feature map:

$$c_i = f\left(W \otimes X_{i:i+k-1} + b\right), \tag{20}$$

where $\otimes$ operator is the summation of elementwise production, b is a bias term and f is a non-linear function. In order to capture complex relation, we use various kernels with different window sizes to extract different features. Then we apply a max-over-time pooling operation over the feature maps to get the final vector:

$$\tilde{c} = max\{c_1, \ldots, c_{m+n-k+1}\}. \tag{21}$$

Finally, a fully connected layer with sigmoid activation is used to compute the probability that the comment is real. The optimization target is to minimize the cross entropy between the ground truth label and the predicted probability.

At the beginning of the adversarial training, we use the maximum likelihood estimation (MLE) to pre-train comment generator on training set until generator reaches convergence. Then we also use MLE to pre-train comment discriminator with real comments and fake comments generated by our generator. After the pre-training, the generator and discriminator are trained alternatively until convergence.

IV. EXPERIMENTS

In this section, we describe our experiments and results on real news comment data. We first introduce a new dataset for this task. Then we compare GANN with several approaches. We use language modeling and human evaluation to evaluate GANN and detail results analysis.

A. DATA SET

The news comment data comes from 163.com, which is one of the most famous news sites in China and has a lot of news comments. We crawl the data from January, 2015 to May, 2017 with a web spider. The data contains detailed information of news, e.g., title, content, and public time, and corresponding comment information, e.g., comment user, comment time and comment text. In our experiments, we only use news titles and corresponding comment texts. In order to reduce noisy data, the comments whose lengths are shorter than 10 and greater than 100 are filtered. Usually the number of comments of each news is greatly different. We filter news whose comment number are less than 50 and for each news we choose 100 comments at most. Then we select the most popular words as the vocabulary and other words are replaced with UNK. Finally, we choose a subset from the filtered data as our experiment dataset which contains 2,000,000 comments and 22,117 news. The average length of comment is about 16 words. The whole data is randomly split into train, validation, and test data according to the ratio 18:1:1.

B. SETUP

We use a one-layer biLSTM encoder and a one-layer LSTM decoder. The dimensions of word embeddings and hidden

TABLE 1. Evaluation results on the test set.

Method	Perplexity
Enc2Dec	196.6
Enc2Dec+ga	168.7
Enc2Dec+ga+rc	162.6
GANN	162.1

vectors are set to 128 and 256 both in the encoder and decoder. The batch size is set as 16. All the parameters are randomly initialized by sampling from a uniform distribution $[-0.02, 0.02]$. We train the model using Adagrad [29] with learning rate 0.15 and an initial accumulator value of 0.1. We also clamp gradient values into the range $[-2.0, 2.0]$ to avoid the exploding gradient problem [30]. The number of epochs is determined by early stopping on the validation set.

C. LANGUAGE MODELING

Perplexity is a common evaluation method of language modeling. The perplexity score measures the probability of the test sentences appearing. The higher the probability is, the lower the perplexity is. The lower perplexity means the model is better.

We describe the comparison methods as follows:

Enc2Dec: This method uses the basic encoder-decoder framework which encodes the news and then decodes to generate comments.

Enc2Dec+ga: Basic model with our proposed gated attention mechanism, which uses context adaptively and selectively.

Enc2Dec+ga+rc: Basic model with our proposed gated attention mechanism and relevance control concurrently.

GANN: Our proposed model with gated attention mechanism, relevance control and comment discriminator.

As shown in Table 1, we compute perplexity scores for these methods. The result of basic encoder-decoder is the worst whereas other methods taking account of contextual information are better. This shows that contextual information is important for generating comments and our proposed gated attention mechanism handles the contextual information efficiently. It improves the performance significantly. Compared to the model just using gated attention mechanism, the model using gated attention and relevance control simultaneously achieves lower perplexity. This demonstrates the relevance control contributes to generating better comments. GANN achieves the best performance through using three mechanisms concurrently. The perplexity drop a little when adding an extra discriminator. We conjecture that in our task each input sequence corresponds to many target comments and there is a big difference between these comments. The discriminator gets lost between these comments. In the following experiments, we use GANN to generate comments.

D. DIVERSITY OF COMMENTS

There are usually two approaches for generation: beam search [24], which is widely used in neural machine translation, and random sample [5]. We tried beam search and find that the comments generated by the beam search are generally very trivial without much variation. When generating comments, we use the random sampling method with the temperature value. The probability controlled by temperature is calculated as follows:

$$\tilde{p}_i = \frac{\exp\left(\frac{p_i}{T}\right)}{\sum_{k=1}^{|V|} \exp\left(\frac{p_k}{T}\right)}. \tag{22}$$

The temperature value is the trade-off between the diversity and the correctness. If the temperature is low, the correctness of the comments is high. If the temperature is high, the diversity is high and the error may be greater.

TABLE 2. Generated comments with different temperatures.

标题：女子扶摔倒老人被质疑摆拍想当网红 Title: Woman helping fallen old man to stand was questioned to pose for being famous	
Temperature	**Generated Comments**
0.2	我觉得，他是个好人， I feel, he is a good man,
0.2	我觉得他也是个好东西， I feel he also is a good guy,
0.5	这就是中国的现状， This is the status of China,
0.5	看了标题，我觉得是扶不扶。 Looking at the title, I think it is to help or not.
0.8	好人，坏人真好人一生平安！ Good man, bad man nice man life safe!
0.8	他不会忘了，早毙了。 He will not forget, early death.

As shown in Table 2, we test the different temperature values with the range of 0 to 1 in order to generate comments with diverse topics. Overall, too low temperature gives common comments and lacks variety. And too high temperature causes some errors. When using low temperature value, the model trends to generate similar comments. For example, "I feel, he..." and "I feel he..." have the same topic and start with the same word. However, we want to generate diversified comments with different topics and forms like natural comments. This indicates that low temperature is not a good choice. When using high temperature value, the model generates diversified comments in topic and form as we want. Nevertheless, there are some grammatical errors or repetitions in generated comments, like "He will not forget, early death." This demonstrates high temperature is also not suitable for generation. The generated comments look better when the temperature is 0.5, which not only generates diversified comments with different topics and forms but also ensures readability. Therefore we use 0.5 in all experiments of comment generation.

In order to evaluate the effect of our proposed relevance control, we generate some comments using different relevance levels. As shown in Table 3, there are two sampled news and corresponding comments generated by different relevance levels. When using 0 level, the model trends to generate general comments, e.g., "I am speechless...", "I would like to know...". The relevance between these comments and

TABLE 3. Generated comments with different relevance.

Relevance	Generated Comments
标题：女子偷上千件快件:习惯性拿一下，又不是到家里偷 Title: Woman stole a thousand pieces of express: it is habit to take, not to steal home	
0	我也是醉了！不知道为什么！ I am speechless! I do not know why!
1	这种人，居然还有那么多快递 Such people, there are actually so much express
2	我还以为是快递员呢，我就想知道还能怎样? I think she is a courier, I want to know what else?
标题：这些钱要涨这些税要减，2017年这些红包请查收 Title: These money to rise and these taxes to be reduced, please check these red envelopes in 2017	
0	我就想知道这是谁的? I would like to know whose this is
1	呵呵，现在的税收都是国家的，没几个人的 Oh, now the tax is for the country, not for us
2	说的好像是降了税一样， It seems like the same as Tax reduction

TABLE 4. Human evaluation.

	TP	FP	TN	FN
number	75.6	39.7	60.3	24.4

news is small. And these comments could correspond to much news. It seems that these comments are unvalued. In fact, they conform to the characteristics of real comments, which means that many users often write comments of this type. This indicates that our 0 level imitates these general comments successfully. When using 1 and 2 level, the generated comments are difficult to distinguish in degree of relevance. These comments talk about some aspects of news and are more relevant to the news. This demonstrates that 1 and 2 level imitates the detailed comments. Overall, the results show that the relevance control we propose is useful to generate diversified comments with different degrees of relevance.

E. HUMAN EVALUATION

In order to evaluate our model we conduct this experiment of human evaluation. We randomly select 100 real comments from the test data set as positive examples. After that, we use our model to generate fake comments as negative examples. The fake comments depend on the same title context as the real comments. Then we get the corresponding 100 fake comments. We shuffle these 200 examples and then ask three graduate students to judge whether the comments are written by real users or not. We summarize the final results into Table 4.

We can see that more than 39% of the fake comments generated by our model are misclassified by the users, and around 75% of the real comments are correctly classified. This illustrates that the comments generated by our model are relatively natural. There are about 25% of the real comments misclassified as fake examples, which indicates that

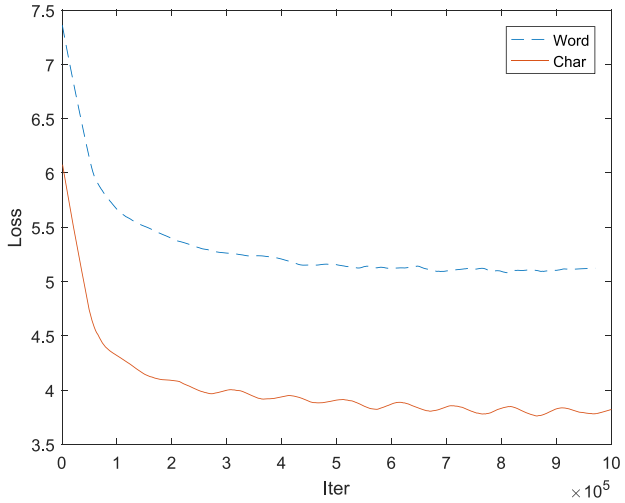


FIGURE 5. Loss curve.

real comments still exit problems. We analyze these comments and find that the majority of these are irrelevant to news content. They talk about completely different topics and some of them even have spelling mistakes. About 60% of the fake comments are correctly classified. We analyze these comments and find that these comments are too general and hardly include favorable information. Compared to these, the comments misclassified as real examples are more relevant to the news context. They often talk about certain aspect of news context. This demonstrates that the relevance between comments and news is the most important factor of human judgment.

F. WORD VS CHARACTER

In all experiments above we use word-level model. We conduct this experiment to compare the word-level and character-level generation. In word-level experiments, we segment sentences to obtain words by extra tool and select the most common words as vocabulary. For the character-level it is easy to obtain characters and all characters are retained as vocabulary about 10,000 characters. Notice that we use a character as "a word" in character-level model and one or more characters as "a word" in word-level model. The other hyper-parameters are the same. We compare the loss curve and generated comments.

As show in Fig. 5, we can see that the two models almost reach convergence in the same iterations. Compared to the word-level model, the character-level model has a small vocabulary size. Therefore it converges to a lower loss value and perplexity of this model is better. In addition, the character-level model spends less time on each iteration than the word-level one and has about three times faster speed. According to these results, the character-level model outperforms the word-level one.

There are sampled comments generated by word and character model depending on the same news context in Table 5. From the title of news we know that the news talks about

TABLE 5. Generated comments with word and character level.

标题：上海一外卖哥骑电瓶车与机动车相撞，抢救无效身亡 Title: Takeaway brother riding battery car was hit by a motor vehicle and died after the rescue	
Type	**Generated Comments**
word	希望外卖小哥一路平安，新年大吉吧 I hope takeaway brother have a pleasant journey, and a happy new year
word	这就是传说中的外卖小哥。 This is the legendary takeaway brother.
char	这个是故意杀人的， This is a deliberate murder,
char	这个不是弱智吗？这是中国人 Is this not mentally handicapped? This is the Chinese people

"takeaway" brother. The comments generated by the word-level model look pretty good, which have the same subject "takeaway" as news context and more informational words, e.g., "a pleasant journey" and "a happy new year". However, the comments generated by the character-level model look more common, which use more general words, e.g., "is" and "people". The word-level model recognizes the word "takeaway", hence the generated comments contain the same subject of news and look like written by human. The character-level model just recognizes the word "take" and "away", thus it generates more general words. This indicates that word-level model outperforms the character-level model for that the comments generated by the word-level are more informational. However, the word-level model has to face the problem of unknown words although using a big vocabulary.

G. DISCUSSION

We conduct a series of experiments to validate our model. Results of language modeling confirm that our gated attention mechanism handles the news context efficiently and we think it can be applied to other context-dependent tasks well. However, the adversarial training improves the performance little. We conjecture that each news corresponds to many comments and the discriminator gets lost between these comments. We will explore the adversarial training for one-to-many data in the future. Furthermore, we made a comparison between word-level generation and character-level generation. The results indicate that both of them have their limits in comment generation. Therefore, combining word representation and character representation may be a great choice.

V. CONCLUSION AND FUTURE WORK

In this paper, we have studied the task of automatic generation of news comments and propose a Gated Attention Neural Network model(GANN) for the task. GANN is built on the encoder-decoder framework and we apply three mechanisms to boost it. The gated attention mechanism deals with news context effectively and improves the performance significantly. Moreover, the relevance control contributes to generating diversified comments. Furthermore, the discriminator boosts GANN by the adversarial training. Experiments on the large dataset show the effectiveness of GANN. The generated news comments are close to human comments.

This work suggests several interesting directions for future research. We will try to integrate other methods like a planning model to generate longer comments such as comment articles. We can also incorporate user information to generate personalized comments. Moreover, we will combine word representation with character representation to improve the performance.

REFERENCES

[1] E. Reiter and R. Dale, *Building Natural Language Generation Systems*. Cambridge, U.K.: Cambridge univ. Press, 2000.

[2] D. L. Chen and R. J. Mooney, "Learning to sportscast: A test of grounded language acquisition," in *Proc. ACM 25th Int. Conf. Mach. Learn.*, 2008, pp. 128–135.

[3] S. G. Sripada, E. Reiter, and I. Davy, "SumTime-Mousam: Configurable marine weather forecast generator," *Exp. Update*, vol. 6, no. 3, pp. 4–10, 2003.

[4] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic question generation for vocabulary assessment," in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process.*, 2005, pp. 819–826.

[5] A. Graves. (Aug. 2013). "Generating sequences with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1308.0850

[6] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 1017–1024.

[7] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. (May 2016). "Generating sentences from a continuous space." [Online]. Available: https://arxiv.org/abs/1511.06349

[8] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, vol. 2. 2010, p. 3.

[9] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. SLT*, vol. 12. Dec. 2012, pp. 234–239.

[10] T. Wang and K. Cho. (2015). "Larger-context language modelling." [Online]. Available: https://arxiv.org/abs/1511.03729

[11] X. Zhang and M. Lapata, "Chinese poetry generation with recurrent neural networks," in *Proc. EMNLP*, 2014, pp. 670–680.

[12] Q. Wang, T. Luo, D. Wang, and C. Xing. (Jun. 2016). "Chinese song iambics generation with neural attention-based model." [Online]. Available: https://arxiv.org/abs/1604.06274

[13] Z. Wang *et al.* (Dec. 2016). "Chinese poetry generation with planning based neural network." [Online]. Available: https://arxiv.org/abs/1610.09889

[14] A. Sordoni *et al.*, "A neural network approach to context-sensitive generation of conversational responses," in *Proc. Assoc. Comput. Linguistics*, 2015, pp. 196–205. [Online]. Available: http://aclweb.org/anthology/N15-1020

[15] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young. (Aug. 2015). "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems." [Online]. Available: https://arxiv.org/abs/1508.01745

[16] J. Tang, Y. Yang, S. Carton, M. Zhang, and Q. Mei. (Nov. 2016). "Context-aware natural language generation with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1611.09900

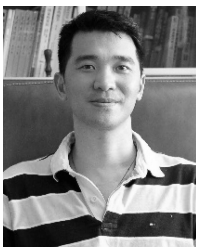
[17] F. Costa, S. Ouyang, P. Dolog, and A. Lawlor. (Jul. 2017). "Automatic generation of natural language explanations." [Online]. Available: https://arxiv.org/abs/1707.01561

[18] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1. 2017, pp. 623–632.

[19] M. A. Walker, O. C. Rambow, and M. Rogati, "Training a sentence planner for spoken dialogue using boosting," *Comput. Speech Lang.*, vol. 16, no. 3, pp. 409–433, 2002.

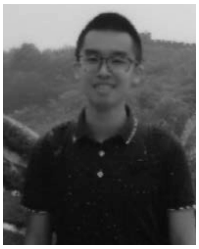
[20] A. Stent, R. Prasad, and M. Walker, "Trainable sentence planning for complex information presentation in spoken dialog systems," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, p. 79.

[21] A. Cheyer and D. Guzzoni, "Method and apparatus for building an intelligent automated assistant," U.S. Patent 8 677 377 B2, Mar. 18, 2014.
[22] A. H. Oh and A. I. Rudnicky, "Stochastic language generation for spoken dialogue systems," in *Proc. Assoc. Comput. Linguistics*, vol. 3. 2000, pp. 27–32. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1117562.1117568
[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[24] D. Bahdanau, K. Cho, and Y. Bengio. (Sep. 2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473
[25] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[26] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.
[27] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI*, 2017, pp. 2852–2858.
[28] X. Zhang and Y. LeCun. (Feb. 2015). "Text understanding from scratch." [Online]. Available: https://arxiv.org/abs/1502.01710
[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[30] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

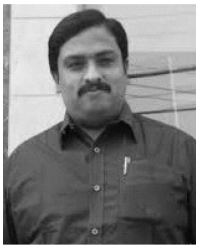


HAI-TAO ZHENG received the bachelor's and master's degrees from the Department of Computer Science, Sun Yat-Sen University, in 2001 and 2004, respectively and the Ph.D. degree in medical informatics, Seoul National University. He is currently an Associate Professor with the Graduate School, Shenzhen, Tsinghua University, Shenzhen, China. He has authored over 30 papers including ten SCI journal papers. His research interests include artificial intelligence, semantic web, information retrieval, machine learning, and medical informatics. He has a deep understanding of the semantic web and knowledge mining fields.

WEI WANG received the B.S. degree in computer science from Yanshan University, Qinhuangdao, China, in 2016. He is currently pursuing the Ph.D. degree with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China. His research interests are artificial intelligence, natural language processing, and machine learning.



WANG CHEN is currently pursuing the master's degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include recommender system, machine learning, and deep learning.



ARUN KUMAR SANGAIAH received the M.E. degree in computer science and engineering from the Government College of Engineering, Anna University, Tirunelveli, India, and the Ph.D. degree in computer science and engineering from the VIT University, Vellore, India. He is currently an Associate Professor with the School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bioinformatics, and embedded systems. He has authored over 100 publications in different journals and conference of national and international repute. His current research work includes global software development, wireless ad hoc and sensor networks, machine learning, cognitive networks, and advances in mobile computing and communications. He has also registered a one Indian patent in the area of computational intelligence. He is responsible for the Editorial Board Member/Associate Editor of various international journals.