Gossiping the Videos: An Embedding-Based Generative Adversarial Framework for Time-Sync Comments Generation

Guangyi Lv[1], Tong Xu[1], Qi Liu[1], Enhong Chen[1(✉)], Weidong He[1], Mingxiao An[1], and Zhongming Chen[2]

[1] Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
gylv@mail.ustc.edu.cn, cheneh@ustc.edu.cn

[2] Quantum Lab, Research Institute of OPPO, Shanghai, China

Abstract. Recent years have witnessed the successful rise of the time-sync "*gossiping comment*", or so-called "**Danmu**" combined with online videos. Along this line, automatic generation of Danmus may attract users with better interactions. However, this task could be extremely challenging due to the difficulties of informal expressions and "semantic gap" between text and videos, as Danmus are usually not straightforward descriptions for the videos, but subjective and diverse expressions. To that end, in this paper, we propose a novel **E**mbedding-based **G**enerative **A**dversarial (**E-GA**) framework to generate time-sync video comments with "gossiping" behavior. Specifically, we first model the informal styles of comments via semantic embedding inspired by variational autoencoders (VAE), and then generate Danmus in a generatively adversarial way to deal with the gap between visual and textual content. Extensive experiments on a large-scale real-world dataset demonstrate the effectiveness of our E-GA framework.

1 Introduction

Recent years have witnessed the booming of the novel time-sync comments on online videos, or so-called "**Danmu**" [10,11], which describes the scene that massive comments flying across the screen just like bullets [14]. This new business mode could not only enrich the video with textual information but also attract viewers with better interactions. For instance, the report of iQiYi[1], a leading Danmu-enabled video-sharing platform in China, revealed that Danmus have improved the online user activities, such as views or comments, even by 100 times. Along this line, administrators are encouraged to improve the loyalty

[1] http://digi.163.com/14/0915/17/A66VE805001618JV.html.

© Springer Nature Switzerland AG 2019
Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11441, pp. 412–424, 2019.
https://doi.org/10.1007/978-3-030-16142-2_32

of users with high-quality Danmus. However, due to the limitation of "grass-root" users, the quantity and quality of Danmu could be hardly ensured. Thus, solutions for automatic Danmu generation is urgently required.

Usually, prior arts conducted the short text generation mainly following the idea of tagging method [25], textual summarization [4,17] or Question-answering system [1]. Nevertheless, though large efforts have been made, these brilliant works may not be suitable for the Danmu generation task due to its unique characters. Indeed, Danmu is not just the objective statement of video content, more importantly, it could be the "**gossiping**" to the video. First, different from the image caption techniques, Danmu always indicates the **subjective** opinions, e.g., "*I like Penny*" and "*Sheldon is so cute*" (from the American TV sitcom "The Big Bang Theory"). Second, the content of Danmus could be more **diverse**, which is not limited to the current episode of video, e.g., we can see "*Bazinga*", the pet phrase of Sheldon, in Danmus at anywhere even without Sheldon. Besides, the expression of Danmus could be informal, as emotions (e.g., "O(∩_∩)O") or slangs (e.g., "*lol*" which means laughing), which could be more **fluent** just like human talking, but cannot be interpreted by literal meanings and thus increase the difficulty of generation.

To that end, in this paper, we propose a novel **E**mbedding-based **G**enerative **A**dversarial framework (**E-GA**) to generate the gossiping Danmus of videos. Specifically, considering the informal expressions in Danmu, we represent both the video scenes and textual information as vectors. Then, to deal with the *semantic gap* between visual content and user opinions, a generative adversarial model is adapted to learn the latent mapping between visual space and semantic space. Along this line, the proper and diverse semantic vectors will be generated, and then decoded as sentences. To the best of our knowledge, we are among the first ones who attempt to generate Danmu-like comments with combining both embedding and adversarial approaches. Extensive experiments on a large-scale real-world dataset demonstrate the effectiveness of our E-GA framework, which validates the potential of our framework on generating "gossiping" text in Danmu-enable social media platforms.

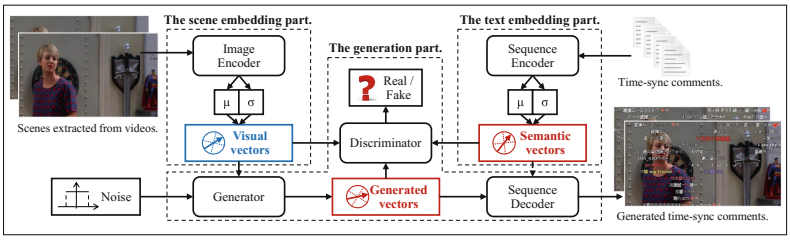


Fig. 1. The overall architecture of the generation framework.

2 Problem Definition and Technical Solution

In this paper, we target at generating Danmus for video frames. Formally, we first give the definition as follow:

Definition 1 (Danmu Generation). *Given the training set of video frames* $\{\boldsymbol{v}_i\}$*, where* $\boldsymbol{v}_i \in \mathcal{V}$ *denotes the i-th frame in video, combined with related Danmus as* $\mathcal{S}_i = \{\boldsymbol{s}_{ik}\}$*. Our target is to learn a Danmu generator* $\mathbb{G}$*, so that a series of Danmu-like comments* $\{\boldsymbol{s}'_{kj}\}$ *could be produced for gossiping any given frame* $\boldsymbol{v}'_i \in \mathcal{V}'$ *in the test set.*

Specifically, as we mentioned above that we target at generating the "gossiping" Danmus for given video frames, we have to satisfy the following three requirements to ensure the gossiping characters:

1. **Relation.** The generated Danmus must be semantically related to the given frame.
2. **Diversity.** The generated Danmus should be more than only the description of the objective truth in the frame. They should be subjective and semantically diverse.
3. **Fluency.** The generated Danmu should be fluency, i.e., their style should be similar to the human-written comments.

Along this line, to satisfy all the three requirements above, we formulate our solution in the following way. First, according to the basic task, i.e., generating a sequence of comments given the video frame, we propose a generator $\mathbb{G}$ to model the probability distribution $P(\boldsymbol{s}|\boldsymbol{v})$. Then, considering the requirements on *semantic relation*, we adopt the Generative Adversarial Networks (GANs) structure [16], and further introduce a noise vector $\boldsymbol{\tau}$, so that the requirements on *diversity* could also be satisfied.

Correspondingly, the generator $\mathbb{G}$ could be re-formulated as $\{\boldsymbol{s}_k\} = \mathbb{G}(\boldsymbol{\tau}|\boldsymbol{v})$. However, here the generated Danmu, as the sequences of words, will be discrete but not continuous as prior arts. Thus, requirements of *fluency* could be unsatisfied with directly using the GAN [28]. Moreover, the informal expressions exist in Danmu may further increase the difficulty in understanding the relations between frames and text. To address these challenges, we design an **E**mbedding-based **G**enerative **A**dversarial framework (E-GA), where the frames $\mathcal{V}$ and comments $\mathcal{S}$ are first represented into low dimensional continuous spaces $\mathcal{H}_v$ and $\mathcal{H}_s$. Then, we further adapt our generator as $\{\boldsymbol{h}_{sk}\} = \mathbb{G}(\boldsymbol{\tau}|\boldsymbol{h}_v)$, in which $\boldsymbol{h}_{sk} \in \mathcal{H}_s$ and $\boldsymbol{h}_v \in \mathcal{H}_v$. Finally, Danmu sentences $\boldsymbol{s}_i$ will be reconstructed from $\boldsymbol{h}_{si}$.

In summary, the overall framework of our E-GA model is illustrated in Fig. 1, which includes two parts, namely (1) the embedding part and (2) the generation part. Technical details will be introduced in the following sections.

2.1 The Embedding Part

First, we will introduce the detail of embedding part. In order to better model the internal relations for the frames and text, we choose to perform data

representation via the Variational AutoEncoders (VAE) [13], which is based on a regularized standard autoencoder. It modifies the conventional ones by using a posterior distribution $q(\boldsymbol{z}|\boldsymbol{x})$ instead of the deterministic embedding $\phi(\boldsymbol{x})$ for input $\boldsymbol{x}$. Reconstruction of $\boldsymbol{x}$ is generated by sampling a vector $\boldsymbol{z}$ from $q(\boldsymbol{z}|\boldsymbol{x})$ and then passing it through a decoder. In addition, to ensure that the embedding space is continuous where any point (vector) can be decoded to a valid sample, the posterior $q(\boldsymbol{z}|\boldsymbol{x})$ is regularized with its KL-divergence from a prior distribution $p(\boldsymbol{z})$, which usually follows standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{1})$. The objective function takes the following form:

$$L = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log\ p(\boldsymbol{x}|\boldsymbol{z})] + D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})), \tag{1}$$

where the expectation term is known as the reconstruction loss L_{rec}, while the other term denotes the KL-loss L_{KL}.

Though the VAE based model can achieve decoding vectors to human acceptable data, e.g., images or fluency sentences, its embedding ability has been largely weakened. Note that, the "embedding ability" here refers to how well the representations can reconstruct their original inputs. For example, if there are embedding vectors $\boldsymbol{h} = \phi(\boldsymbol{x})$ that can be decoded to the inputs $\boldsymbol{x}$ with little loss, we normally say that ϕ have good embedding ability. In contrast, if a series of representations fail to reconstruct the original inputs, there is definitely a loss of information. At the same time, the associated reconstruction loss L_{rec} will be large. Thus, we are not going to use the VAE directly. Considering the KL term in Eq. 1, the KL divergence for diagonal Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ can be formulated by:

$$L_{KL} = \sum_{i=1}^{N}(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1), \tag{2}$$

which is composed of the "μ-term" and the "σ-term". As we know, for a converged VAE, these two terms will ideally set $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ to $\mathbf{0}$ and $\mathbf{1}$ respectively, which will result in poor embedding effect. In our task, both of the ability of embedding and decoding are needed. On one hand, we need the proper representations $\boldsymbol{\mu}$ to feed into the generator. On the other hand, we also need the decoder to generate new sentences from $\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ rather than giving the existing sentences from the training set. To this end, we loose the KL constraint by replacing the μ-term with $\max(\mu_i^2 - \mu_0^2, 0)$:

$$L_{KL} = \sum_{i=1}^{N}(\max(\mu_i^2 - \mu_0^2, 0) + \sigma_i^2 - \log(\sigma_i^2) - 1), \tag{3}$$

so as to $\boldsymbol{\sigma}$ still converge to $\mathbf{1}$ while μ_i can be in the range of $[-\mu_0, \mu_0]$. Further, to measure the embedding capacity for the modified model, we define a metric as follows:

$$C = \mathbb{E}_{\mu_i \sim \mathcal{U}(-\mu_0, \mu_0)}\left[\frac{D_{KL}\big(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\big)}{H\big(q(\boldsymbol{z}|\boldsymbol{x}), p(\boldsymbol{z})\big)}\right] = 1 - \frac{\sqrt{\ln 2\pi \mathrm{e}}}{\mu_0}\ \arctan\frac{\mu_0}{\sqrt{\ln 2\pi \mathrm{e}}}, \tag{4}$$

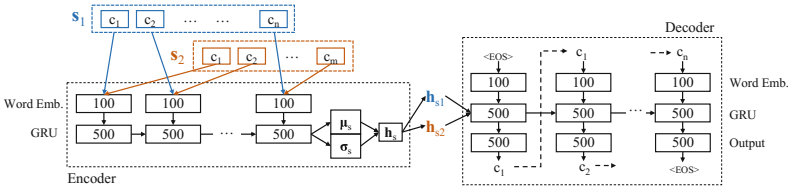


Fig. 2. The RNN structure of sentence encoder and decoder. Size of each layer is labeled on the box. Note that the encoder and decoder share the same parameters for word embedding layer.

where H denotes the cross entropy of the two distribution. C is valued in $[0, 1)$, and we could balance the effect of embedding and decoding by tuning μ_0 based on this. We will discuss more about this later in Sect. 3.4.

Next, to be specific, for video frames, we set up an encoder ϕ_v to encode an image $\boldsymbol{v} \in \mathcal{V}$ as a posterior distribution $q(\boldsymbol{h}_v|\boldsymbol{v})$. Typically, we use a diagonal Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v^2)$ to present this posterior, where $(\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v) = \phi_v(\boldsymbol{v})$. Then, to formulate the loss function and learn the model, a visual vector $\boldsymbol{h}_v$ is sampled from $q(\boldsymbol{h}_v|\boldsymbol{v})$ and then sent to a decoder ψ_v. The image is finally reconstructed as $\boldsymbol{v}' = \psi_v(\boldsymbol{h}_v)$. The reconstruction loss is in the form of Mean Squared Error (MSE):

$$L_{rec} = \frac{1}{N} \sum (\boldsymbol{v}' - \boldsymbol{v})^2. \tag{5}$$

Specially, the encoder ϕ_v and decoder ψ_v are implemented by deep convolutional networks with 4 layers as used in [19].

For Danmu sentences, the situation is a little different. We design character level Gated Recurrent Unit (GRU) [5] networks as encoder ϕ_s and decoder ψ_s, as shown in Fig. 2. At each time, a pair of sentences $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ that are selected from the same frame are first put into the encoder by characters to get their posterior distributions $\mathcal{N}(\boldsymbol{\mu}_{s1}, \boldsymbol{\sigma}_{s1}^2)$ and $\mathcal{N}(\boldsymbol{\mu}_{s2}, \boldsymbol{\sigma}_{s2}^2)$. Like frame embedding, $\boldsymbol{h}_{s1}$ and $\boldsymbol{h}_{s2}$ which are sampled from the two distributions are put into the decoder. In the decoder, for every single sentence, the corresponding reconstruction loss is the sum of the negative log likelihood of the correct character at each step:

$$L_{rec}(\boldsymbol{s}) = -\log\ P(\boldsymbol{s}|\boldsymbol{h}_s) = -\sum_{t=1}^{N} \log\ P(c_t|\boldsymbol{h}_s, c_0, ..., c_{t-1}). \tag{6}$$

More importantly, to model the deeper semantic meaning of Danmus, we also involve a semantic loss formulated as:

$$L_{sem}(\boldsymbol{s}_1, \boldsymbol{s}_2) = \text{dist}(\boldsymbol{\mu}_{s1}, \boldsymbol{\mu}_{s2}), \tag{7}$$

in which we take the assumption of "temporal correlation" [14], i.e., comments appear in the same frame hold the similar topics (relevant to the frame, but semantically diverse). Here we choose cosine distance as the distance function dist(). Finally, the overall reconstruction loss for Danmu embedding is given by:

$$L_{rec} = L_{rec}(\boldsymbol{s}_1) + L_{rec}(\boldsymbol{s}_2) + L_{sem}(\boldsymbol{s}_1, \boldsymbol{s}_2). \tag{8}$$

2.2 The Generation Part

In the generation part, we set up Conditional Generative Adversarial Model which consists of two "adversarial" models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. Here, since we aim to produce semantic vectors from the visual vectors, both G and D are implemented by deep neural networks.

In detail, we choose to utilize our GAN as a Wasserstein GAN [2]. For G, the visual vector $\boldsymbol{h}_v$ and the noise vector $\boldsymbol{\tau}$ are first concatenated, and then put into the hidden layers with size 1000 and 500. Here, we perform the batch normalization [12] for every layer to reduce the internal-covariate-shift by normalizing its input distributions to the standard Gaussian distribution, and leaky ReLU with leak value 0.01 is used as the activation function. Then, a linear transformation is took place on the output to produce the "fake" semantic vector, i.e., $\boldsymbol{h}_s = G(\boldsymbol{\tau}|\boldsymbol{h}_v)$.

Similarly, for D, the input is the concatenation of a visual vector $\boldsymbol{h}_v$ and a (fake) semantic vector $\boldsymbol{h}_s$, while the hidden layers are sized as 2000 and 1000 with the same activation function. Please note that batch-norm should not be used for a discriminator since it can cause the model unable to converge. Finally, the critical output $y = D(\boldsymbol{h}_s|\boldsymbol{h}_v)$ is calculated by linearly mapping the hidden state to a scalar, which indicates whether the input semantic vector is fake or not. Furthermore, G and D are trained alternatively and the objective function of a two-player min-max game would be:

$$\min_G \max_D V(D,G) = \mathbb{E}_{p(\boldsymbol{h}_s|\boldsymbol{h}_v)}[D(\boldsymbol{h}_s|\boldsymbol{h}_v)] - \mathbb{E}_{p(\boldsymbol{\tau})}[D(G(\boldsymbol{\tau}|\boldsymbol{h}_v)|\boldsymbol{h}_v)].$$

2.3 Learning the Model

We then turn to introduce details about learning the model. With recalling the Fig. 1, the training process can be divided into two stages: (1) We separately learning the two autoencoders with the frames and comments from the videos. After the parameters are fine-tuned, we store the models including the image encoder ϕ_v, sequence encoder ϕ_s and the sequence decoder ψ_s for further use. (2) Based on the autoencoders, we train the generator $\mathbb{G}$ and the discriminator D in a generative adversarial way. Note that in this stage, parameters of ϕ_v, ϕ_s and ψ_s are kept unchanged, only $\mathbb{G}$ and D are updated.

To be specific, in both of the two stages, mini-batch gradient descent is used to optimize the models, where the batch size in our case is 32. For the autoencoders, we use SGD with momentum, where the learning rate and momentum are separately set as 0.1 and 0.6, and at the same time, gradient clipping is performed to constrain the L2 norm of the global gradients not larger than 1.0. To our pilot study, it is crucial to clip the gradients for most of the optimizing algorithms due to the exploding gradients problem even with a very small learning

rate. For the GAN part, we take the RMSProp[2] algorithm with learning rate 10^{-5} and decay 0.9.

Another problem is the trade-off between reconstruction loss L_{rec} and KL-loss L_{KL} when training the embedding models. For a VAE-based model, directly minimizing $L_{rec} + L_{KL}$ may fail to encode useful information [3] in the embedding vector, since in most cases, L_{KL} is far more easy to be optimized, which will yield models that consistently set $Q(z|x)$ equal to $P(z)$. Thus, in our case, we design a simple annealing approach, in which $L_{rec} + \alpha L_{KL}$ is used to replace the original loss function, where α is initialized with 0 and then gradually increased to 1.

3 Experiments

3.1 Data Preparation

We choose to validate our work on a real-world dataset extracted from Bilibili, which is one of the largest video-sharing platforms in China. Specially, totally 2,716 individual movies are extracted, which last for 232,485 minutes and contain 9,661,369 Danmus. To get scene images, we split the videos into frames for every one second.

Since the total number of the frames is too large, key frame extraction is carried out to eliminate the duplicated ones. First, we extract features for frames by constructing the scalable color descriptors (SCD) [15]. Then, based on these features, an affinity propagation algorithm is performed to cluster the frames, and the kernels are collected as our key frames. In our experiment, we got 214,953 key frames with their corresponding Danmus. 80% of them are used as training data, while others for testing.

3.2 Experimental Setup

Baseline. As far as we know, few works about Danmu generation have been done before and there can be mainly three kinds of models for generation tasks. Thus, to evaluate our model, we consider the corresponding straightforward baseline models to compare with.

(1) Encoder-Decoder framework. We train a Convolutional Neural Network (CNN) as the encoder to get the representations of frames. The representations are then treated as inputs for a decoder implemented by a Recurrent Neural Network (RNN). The model is similar to the Neural Image Caption [22].

(2) Conditional Variational Autoencoders (CVAE). The CVAE [21] is based on traditional VAE which has an condition input $\boldsymbol{y}$ to both encoder and decoder. In our experiment, we take the representations of the frames as $\boldsymbol{y}$.

[2] http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

(3) Simple Generative Adversarial Networks. Similar with CVAE, generative adversarial nets can be extended to a conditional model [16]. We can perform the conditioning by feeding extra information $\boldsymbol{y}$ (the representations of frames in our experiment) into both the discriminator and generator.

Artificial Judgement. Since heuristic rules could hardly judge whether a sentence should be "gossiping" of a given video, to evaluate the Danmu generation models, a human study is carried out, in which we have 40 experts who have years of experience in watching Danmu-enabled videos. While, as the amount of all generated Danmu is really huge for humans, we also developed a web-based GUI for online labeling. For each time a person logs in the system, 20 video frames are randomly sampled from the test set with their corresponding generated Danmus. Then he/she is asked to click the Danmus which are thought as fake. Our system will then label the clicked ones as "fake", and the others as "escaped". We evaluate the models based on the percentage of the "escaped" Danmus, which we call it "*Human Recall*".

Metrics. To eliminate the errors caused by the human raters, we will take metrics which can be automatically computed as our alternative measurements. The *BLEU* score [18] which is a form of precision of word n-grams between generated and reference sentences has been commonly used in machine translation and image description. In this paper, we use the character level BLEU-4 score to measure the overall performance. The references set of BLEU are 3 sentences randomly selected from the existing comments of the corresponding frame. Additionally, we also define *Fluency* and *Diversity* metrics to measure the performances on multiple aspects. In detail, for each Danmu sentence $\boldsymbol{s}$, we split it into n-gram tokens $t \in \mathcal{T}_{\boldsymbol{s}}$. The Fluency and Diversity are separately defined in the form below:

$$Fluency = \frac{\sum_{t \in \mathcal{T}_{\boldsymbol{s}}} [t \in \mathcal{T}] \text{len}(t)}{\sum_{t \in \mathcal{T}_{\boldsymbol{s}}} \text{len}(t)}, \quad Diversity = 1 - \frac{1}{N} \sum_{\boldsymbol{s}_i, \boldsymbol{s}_j \in \mathcal{S}', i \neq j} \frac{2|\mathcal{T}_{\boldsymbol{s}_i} \cap \mathcal{T}_{\boldsymbol{s}_j}|}{|\mathcal{T}_{\boldsymbol{s}_i}| + |\mathcal{T}_{\boldsymbol{s}_j}|},$$

where $\mathcal{T}$ denotes the n-gram tokens for all human written sentences in the train set, $[t \in \mathcal{T}]$ is an indicator function whose value is 1 if $t \in \mathcal{T}$ otherwise is 0, and $\mathcal{S}'$ indicates all sentences generated for the same scene and N is the total number of the pair combinations in $\mathcal{S}'$.

Table 1. Performance of these models.

	Human Recall	BLEU-4	Fluency	Diversity
Encoder-Decoder	0.4572	0.168214	0.678827	0.904757
Conditional VAE	0.5580	0.174298	0.733117	0.948959
Simple-GAN	0.3454	0.129924	0.440087	0.705946
E-GA	**0.6274**	**0.177638**	**0.845072**	**0.964757**

3.3 Overall Results

The overall experimental results are summarized in Table 1. We can see that our proposed framework outperforms the other models in all of four metrics. Not surprisingly, all models except Simple-GAN achieve high performance on *Fluency*, since for those straightforward RNN-based models, it is easier to imitate human style languages, while simple implemented GAN fails due to the discreteness output in the task. However, all of these methods perform poor on *BLEU*, which we think is also reasonable since our task is quite different from those like translation or image description. As mentioned in Sect. 1, the Danmu senders do not aim to reveal the objective truth in most cases, so the existing Danmus cannot be considered as the only deterministic ground-truth in our experiment. Consequently, it is very difficult and sometimes no need to hit the existing Danmus precisely.

At the same time, we have observed that our model outperforms the others with significant margin on *Human-Recall* and *Diversity* due to the excellent generative ability of GAN. Thus, it is proved to be reasonable that the combination of embedding method and GAN is suitable for Danmu generation task. On one hand, the embedding technology simplifies the GAN structure into DNNs which are more easy to learn. On the other hand, it avoids the discrete problem when training a GAN in generating sequential data.

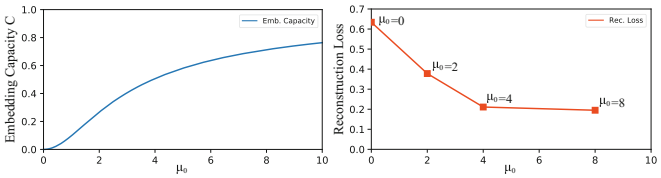
3.4 Balance for Embedding and Decoding Capacity

The performance of our framework can be affected by the embedding/decoding capacity of autoencoders, therefore, it is crucial for us to determine the associated parameter and also necessary to analyze the impacts of them. As mentioned in Sect. 2.1, the embedding effect of a VAE model is naturally opposite to its decoding ability, and thus we involved parameter μ_0 in making a trade-off. According to Eq. 4, there is a curve that the embedding capacity C changes along with μ_0. As shown in Fig. 3, C is zero at the beginning, which means the model is almost unable to perform sentence representation but perfect in generating. Then, as μ_0 increases, C grows rapidly, and at the same time, the embedding ability will become stronger. As μ_0 continues to become larger, the enhancement for embedding quality is getting less stark.

We examined this by setting up several autoencoders with different μ_0. Here, Table 2 gives some examples with μ_0 set to 0, 2, 4 and 8, and Fig. 3 shows the reconstruction loss changing with μ_0. For every case, three sentences are listed which separately indicate the "input", the "reconstruction" from $\boldsymbol{\mu}$ and the "generation" from a sample from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Obviously, when μ_0 is zero, we got the best generation effect, however, we could hardly reconstruct the original sentence from its representation $\boldsymbol{\mu}$. Then, we can see for μ_0 valued 2 and 4, the reconstructed sentences are much better and the generated ones are still acceptable. At last, if μ_0 is much larger, the reconstruction quality reached the best, while the generated sentence became unreadable for humans. In summary, the results prove that our modification for VAE is reasonable, and in most cases, we can set μ_0 to around 2.

Table 2. Samples from trained autoencoders.

	$\mu_0 = 0, C = 0.0$	$\mu_0 = 2, C = 0.2665$
INPUT	雪人保持的真好	卧槽 双击666啊！
	(The snowman keeps well.)	(OMG Double click 666 AH!)
RECO.	好人的冲击力	卧槽 6666啊
	(The impact of a good man.)	(OMG! 6666 AH!)
GEN.	高能预警QAQ	好可爱啊(oV$_i$)！ ！ ！
	(Warning! High energy! QAQ)	(So cute it is (oV$_i$) !!!)
	$\mu_0 = 4, C = 0.5063$	$\mu_0 = 8, C = 0.7129$
INPUT	绝对不能杀主角	他被亲了我好难受啊啊
	(You can't kill the hero.)	(I feel so upset he was kissed.)
RECO.	绝对不是主角光环	他被亲了我好难受啊啊
	(Definitely not the halo of the hero.)	(I feel so upset he was kissed.)
GEN.	好喜欢你，这首歌神的	黑暗示就好玩的
	(Really like you, the god of the song's)	(The dark indicates good things to play.)



(a) (b)

Fig. 3. The embedding capacity (a) and reconstruction loss (b) w.r.t μ_0.

3.5 Case Study

At last, some typical scene images and the generated Danmus can be seen in Fig. 4. Row 1 and Row 2 are good and bad cases generated by our E-AG framework. Row 3 shows outputs from other baselines. For scenes in the first row, the generated Danmus are mainly focused on expressing viewers' different opinions on the frame, which have very high diversity. Especially, for the scene from row 1 column 2, we can easily recognize it as scared shot. Just like human viewers, our model not only generates Danmus to indicate "the ghost will come", but also send something like "BGM is lovely", "It is an interesting movie" to embolden themselves. Of course, we have to admit that there are also some Danmus do not fit the given scenes. While, to our further observation, we found that most of the miss-generated scenes are images with some strange content. Finally, for some results in the third row, we can hardly imagine the relationship between some of the comments and frames. In summary, the results are interesting, and furthermore, we could intuitively feel the diversity and the gossiping behavior of Danmu-enabled videos.

Fig. 4. Typical cases of generated Danmus. The Chinese sentences are translated into English.

4 Related Work

In this section, we will summarize the prior arts on three related topics, namely *Text Generation*, *Unsupervised Autoencoders* and the *Generative Adversarial Networks*.

Text Generation. Since we have witnessed only a few prior arts which focus on the Danmu analysis, especially for the Danmu generation, we will summarize related works on similar topics with Danmu-like Text Generation, i.e., the Image Caption which focus on extracts "meaningful" descriptions for images. Traditionally, early approaches rely on recognizing the visual elements, and then performing template model, n-gram model, or statistical machine translation to get sentences [8,20]. Recently, end-to-end methods [22,24] are proposed to combine deep convolutional networks and recurrent neural networks as autoregressive models. However, image caption techniques mainly focus on describing the objective facts, which is different from the task the Danmu generation who targets at expressing the subjective opinion of viewers.

Unsupervised Autoencoders. These NN-based techniques are designed for efficient embedding, with the aim of learning an encoder $\phi(\boldsymbol{x})$ by maximizing the likelihood of a probabilistic decoder $P(\boldsymbol{x}|\phi(\boldsymbol{x}))$. Though autoencoders have seen success in pre-training image [23] and sequence [6] models, they may not be effective at extracting for global semantic features, e.g., generating data from the continuous space. In contrast, recently, a variant method called Variational Autoencoder (VAE) [13] has become more widely used for learning generative models. The VAE learns representations not as single points, but as a distribution in the latent space, forcing them to fill the space rather than memorizing

the training data as the isolated vectors. However, according to the features above, the VAE may not be suitable for embedding, due to the difficulty in reconstructing samples from the indeterministic representations.

Generative Adversarial Networks. GANs are methods to generate synthetic data with similar statistical properties as the real one [9]. Instead of explicitly defining a loss from a target distribution, GANs train a generator by receiving a loss from a discriminator which tries to differentiate between real and generated data. Though GANs and its variants have shown great success in Computer Vision domain [7,19], there are still challenges in applying them to the traditional NLP tasks [26–28].

5 Conclusion

In this paper, we proposed an embedding-based framework to generate Danmu-like comments for video scenes. In detail, we first represented key frames and comments into continuous spaces, and then learned the mapping between the two spaces via a generative adversarial approach. Along this line, the proper and diverse semantic vectors will be generated, and then decoded as sentences. Experiments on a real-world dataset showed the potential of our framework on generating "gossiping" text in Danmu-enable social media platforms. In the future, we will improve our framework with more comprehensive factors (e.g., positions, colors) which may help to better understand the meaning.

Acknowledgments. This research was partially supported by grants from the National Natural Science Foundation of China (Grant No. 61727809, U1605251, 61672483, and 61703386), the Anhui Provincial Natural Science Foundation (Grant No. 1708085QF140), and the Fundamental Research Funds for the Central Universities (Grant No. WK2150110006).

References

1. Alupului, M., Ames, A.L., Collopy, B.A.M., Pesot, J.F., Pierce, R., Steinmetz, D.C.: Question-answering system. US Patent App. 15/229,361, 5 August 2016
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017)
3. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)
4. Chua, F.C.T., Asur, S.: Automatic summarization of events from social media. In: ICWSM (2013)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR (2014)
6. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: NIPS, pp. 3079–3087 (2015)
7. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a Laplacian pyramid of adversarial networks. In: NIPS, pp. 1486–1494 (2015)

8. Farhadi, A., et al.: Every picture tells a story: generating sentences from images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 15–29. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_2
9. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
10. He, M., Ge, Y., Chen, E., Liu, Q., Wang, X.: Exploring the emerging type of comment for online videos: Danmu. ACM Trans. Web (TWEB) **12**(1), 1 (2018)
11. He, M., Ge, Y., Wu, L., Chen, E., Tan, C.: Predicting the popularity of *DanMu*-enabled videos: a multi-factor view. In: Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H. (eds.) DASFAA 2016. LNCS, vol. 9643, pp. 351–366. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32049-6_22
12. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
14. Lv, G., Xu, T., Chen, E., Liu, Q., Zheng, Y.: Reading the videos: temporal labeling for crowdsourced time-sync videos based on semantic embedding. In: AAAI, pp. 3000–3006 (2016)
15. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. IEEE TCSVT **11**(6), 703–715 (2001)
16. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
17. Neto, J.L., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization using a machine learning approach. In: Bittencourt, G., Ramalho, G.L. (eds.) SBIA 2002. LNCS (LNAI), vol. 2507, pp. 205–215. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36127-8_20
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
20. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: ICCV, pp. 433–440 (2013)
21. Sohn, K., Yan, X., Lee, H.: Learning structured output representation using deep conditional generative models. In: NIPS, pp. 3483–3491 (2015)
22. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: CVPR, pp. 3156–3164 (2015)
23. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: NIPS, pp. 809–817 (2013)
24. Wang, Z., et al.: Chinese poetry generation with planning based neural network. COLING (2016)
25. Wu, B., Zhong, E., Tan, B., Horner, A., Yang, Q.: Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In: SIGKDD, pp. 721–730. ACM (2014)
26. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: sequence generative adversarial nets with policy gradient. In: AAAI (2017)
27. Zhang, K., et al.: Image-enhanced multi-level sentence representation net for natural language inference. In: ICDM, pp. 747–756 (2018)
28. Zhang, Y., Gan, Z., Carin, L.: Generating text via adversarial training (2016)