Pattern Recognition 111 (2021) 107702

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Enhancing the alignment between target words and corresponding frames for video captioning

Yunbin Tu [a,b], Chang Zhou [c], Junjun Guo [a,b], Shengxiang Gao [a,b], Zhengtao Yu [a,b,*]

[a] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, Kunming 650500, PR China
[b] Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Yunnan, Kunming, 650500, P.R. China
[c] Department of information science and technology, Tsinghua Shenzhen International Graduate School, Guangdong, Shenzhen 518000, PR China

ARTICLE INFO

Article history:
Received 15 April 2020
Revised 27 August 2020
Accepted 13 October 2020
Available online 14 October 2020

Keywords:
Video captioning
Alignment
Visual tags
Textual-temporal attention

ABSTRACT

Video captioning aims at translating from a sequence of video frames into a sequence of words with the encoder-decoder framework. Hence, it is critical to align these two different sequences. Most existing methods exploit soft-attention (temporal attention) mechanism to align target words with corresponding frames, where the relevance of them merely depends on the previously generated words (i.e., language context). As we know, however, there is an inherent gap between vision and language, and most of the words in a caption belong to non-visual words (e.g. “a”, “is”, and “in”). Hence, merely with the guidance of the language context, existing temporal attention-based methods cannot exactly align target words with corresponding frames. In order to address this problem, we first introduce pre-detected visual tags from the video to bridge the gap between vision and language. The reason is that visual tags not only belong to textual modality, but also can convey visual information. Then, we present a Textual-Temporal Attention Model (TTA) to exactly align the target words with corresponding frames. The experimental results show that our proposed method outperforms the state-of-the-art methods on two well known datasets, i.e., MSVD and MSR-VTT. [1]

1. Introduction

The goal of video captioning is to automatically generate a natural language sentence that summarizes the content of a video. The research topic is interesting because not only does it has important practical applications, such as video title generation, blind navigation, and content-based video retrieval, but also it connects Computer Vision with Natural Language Processing which are two major directions in Artificial Intelligence.

Thanks to the rapid development of deep neural network, the encoder-decoder neural network has achieved encouraging performances for video captioning. Specifically, an encoder convolutional neural network (CNN) [1,2] encodes video frames into feature vectors and then a decoder recurrent neural network (RNN) [3] decodes them into a natural language sentence. With this framework, the goal of video captioning [4–6] is to directly translate from a sequence of frames to a sequence of words. To this end, it is critical to build the alignment between these two sequences, for the order and length of them may different. Inspired by the soft-alignment approach [7] which learns to align and translate jointly in neural machine translation (NMT), Yao et al. [4] proposed to incorporate this mechanism, namely the temporal attention mechanism into the encoder-decoder framework and demonstrated its benefits. At time step t, the temporal attention mechanism is capable of deciding which a subset of frames to attend to, making it possible for the decoder to generate each target word only based on its relevant frame, where the relevance is measured by the previously generated words (i.e., language context). As we know, however, there is an inherent gap between vision and language, and most of the words in a caption belong to non-visual words (e.g. “a”, “is”, and “in”). In this situation, merely with the guidance of the language context, the temporal attention mechanism often fails to determine which frames are more relevant to the target words. In other words, temporal attention-based methods merely build coarse alignment between target words and corresponding frames, and thus generate ambiguous descriptions, such as details missing or recognition error. For instance, in Fig. 1, the key object “bird” is missing in (a), and the “girl” is wrongly recognized as the “boy” in (b).

In Fig. 1(a), although the object “bird” occurs in every frame, it is more apparent in the last two frames. Furthermore, in the

* Corresponding author.
E-mail address: ztyu@hotmail.com (Z. Yu).

[1] Our code is available at https://github.com/tuyunbin/Enhancing-the-Alignment-between-Target-Words-and-Corresponding-Frames-for-Video-Captioning

https://doi.org/10.1016/j.patcog.2020.107702
0031-3203/

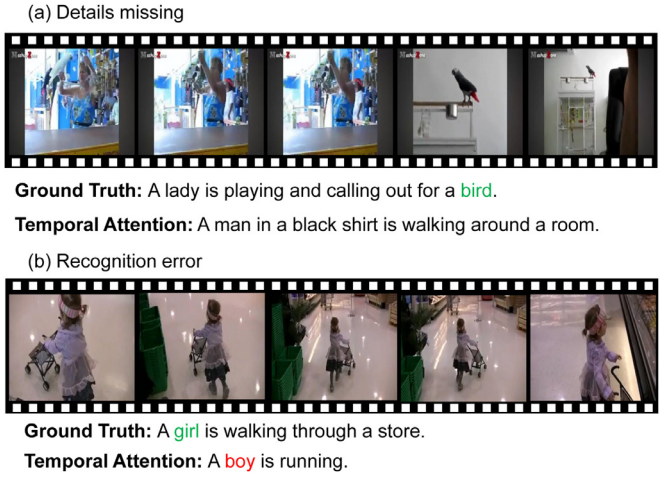

Fig. 1. Two unaligned examples generated by the temporal attention-based method. (a) The key object "bird" is missing. (b) The "girl" is wrongly recognized as the "boy".

ground truth of (a), we can clearly see that most of the generated words belong to non-visual words (i.e., "a", "is", "and", "out", and "for"). Similarly, in Fig. 1(b), the object "girl" is not in a good position in the most frames and the previous word is only "a". In above cases, the temporal attention-based method cannot get sufficient visual cues to select more proper snippets and thus failed to align the target words "bird" and "girl" with the most relevant frames. In order to mitigate this problem, we consider introducing some useful visual cues as "bridges" to help the temporal attention mechanism to build exact alignment between target words and corresponding frames.

To achieve this goal, in this paper we first propose to exploit detected visual tags in frames as "bridges" between vision and language. The reason is that these tags are visual words which not only belong to textual information, but also are capable of conveying visual information. Then, to exploit these tags to enhance the alignment during temporal attention process, we devise a Textual-Temporal Attention Model (TTA) which extends the conventional temporal attention model via adding another textual attention model. Specifically, first, we adopt a CNN encoder to extract frame features for a video. Second, in order to get useful visual tags and inspired by the task of instance segmentation, we opt for a Region-based CNN (R-CNN) to detect each frame and select top-n tags representing the most frequent objects in the video. Finally, we incorporate the proposed Textual-Temporal Attention Model (TTA) into an LSTM decoder to generate target words. The textual attention model is first utilized to select the key visual tags via the language context, because both of them belong to textual modality. Then, the temporal attention model will determine the key frames with the help of the language context plus the key visual tags. Through this manner, the attended frames will be more relevant to target words. Thereinto, not only do key visual tags offer important visual cues to temporal attention process, but also capture fine-grained information that may be neglected by frame features.

In summary, our contributions of this work are as follows:

- Visual tags are introduced to bridge the gap between vision and language.
- A textual-temporal attention model is devised and incorporated into the decoder to build the exact alignment between target words and corresponding frames.
- Extensive experiments on two well-known datasets, i.e., MSVD and MSR-VTT, demonstrate that our proposed approach achieves remarkable improvements over the state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we first review some existing studies. In Section 3, we introduce the overall framework of our proposed method and elaborate each component in the framework. In Section 4, we elaborate the experimental procedure, quantitative analysis, and qualitative analysis. In Section 5, we draw our conclusion and make discussions on this work.

2. Related work

In this section, we provide relevant research progress of previous works in video captioning, the methods exploring the alignment between vision and language and the methods exploiting visual tags.

2.1. Video captioning

To date, the task of video captioning has achieved marked progress in the community of multi-modalities learning. Previous works in this task could be classified into three dimensions, that is, (1) template-based methods, (2) CNN and RNN-based methods and (3) CNN and Transformer-based methods.

Template-based methods Template-based methods [8,9] first predicts a set of visual concepts (e.g., objects, relationships, and attributes) by different classification approaches. Then, a pre-defined sentence template is exploited to form these visual concepts into a caption based on the basic grammar (e.g., subjects-verbs-objects). This kind of approach is intuitive and easy to understanding, but need to cope with the complex data. Moreover, flexible and meaningful captions cannot be generated due to the limitation of predefined templates.

CNN and RNN-based methods The CNN and RNN-based encoder-decoder framework has been widely used in the task of video captioning during these years. Generally, the deep convolutional neural network (CNN) is often opted as the encoder, because it can represent image content with the high-level semantic feature vector. Since the recurrent neural network (RNN) is able to effectively model the sequence generation task, it is often used as the decoder to map a sequence of video frames to a sequence of words. To be more specific, a pre-trained CNN reads a sequence of video frames and outputs corresponding frame features, which in turn are fed into a decoder RNN to generate a caption that summarizes the main content of this video. For instance, Venugopalan et al.[10] first introduced the encoder-decoder framework for video captioning. In their work, they used the mean pooling representation over all frames and fed it into the decoder to generate a description. To alleviate the vanishing gradient problem, they adopted the Long Short-Term Memory (LSTM) [11] as the decoder. Compared to previous template-based methods, encoder-decoder neural network-based methods have made great progress on standard metrics. However, simply averaging all of the frame features often results in the confused representation for video content, because this strategy makes different objects along the temporal sequence fuse disorderly. In order to exploit the temporal structure underlying the video, on one hand, Venugopalan et al.[12] proposed a model, S2VT, to only use an LSTM to encode and decode frame features. Specifically, in the encoding stage, each frame feature was orderly fed into an LSTM to model the temporal dependencies among them. In the decoding stage, the last hidden state of the encoding stage was fed into every time step of the remaining LSTM to generate a caption. On the other hand, Yao et al. [4] presented a TA model which incorporated the temporal attention mechanism into the encoder-decoder framework to selectively focus on a sub-

set of frames for generating each word. Since then, these two models has been two classic baselines for video captioning.

CNN and Transformer-based methods Recently, in the domain of neural machine translation, a new and effective model, namely Transformer [13], has been proposed to replace RNN and has achieved encouraging performances. Inspired by it, Chen et al. [14] has introduced this framework for video captioning. In their work, a sequence of video frames were first fed into both 2-D and 3-D CNNs which then outputted corresponding frame and motion features, respectively. Then, instead of the use of RNN, they attempted to utilize the transformer network for sequential representation and devise two types of fusion blocks in decoder layers for combining different modalities effectively. Similarly, in the task of dense video captioning which aims at both localizing and describing all events in a video, Zhou et al. [15] also proposed to use an end-to-end transformer model for both detecting and describing events in a video. All in all, the use of the transformer provides a new research direction for video captioning and thus is worth deeply exploring in the future.

2.2. Building the alignment between vision and language

In order to generate accurate captions, it is a critical problem for researchers to build exact alignment between vision and language. There are main two directions for addressing this problem. One is the embedding-based method. The other is the attention-based method.

Embedding-based alignment This kind of method aims to build a global alignment by embedding the whole video feature and the sentence feature of a video into a common latent space. Specifically, Pan et al. [16] first performed mean-pooling process over all frame features to get a single video feature. Then, they calculated term frequency (TF) weights over all encoded words to construct the integrated sentence feature. Finally, they embedded the video and the sentence feature into a common space and computed the distance between them. In the training phase, they minimized this distance to enhance their alignment. Guo et al. [17] proposed a similar strategy to enforce the consistency between the sentence feature and the video feature. The slight difference is that their sentence feature was produced by mean-pooling strategy as well. However, the alignment in their work was very coarse, because it built upon the whole video and sentence.

Attention-based alignment This kind of method is based on the attention mechanism, which explored how to build the local alignment between each target word and its corresponding visual and/or semantic feature in a video. Yao et al. [4] first devised a temporal attention model for video captioning, where each target word was generated via the most relevant frames. The relevance was measured by the language context which summarizes all the previously generated words. Tu et al. [18] proposed a spatial-temporal attention model. Given the language context, the spatial attention model first attended to the key regions in each frame, and then focused on the important temporal segments via the temporal attention model. Hori et al. [19] employed two kinds of visual features, that is, frame features extracted from a single frame and motion features extracted from consecutive frames. And they used a multi-modalities attention model to align different kinds of visual features with corresponding words, i.e., frame features are utilized to predict nouns and verbs are predicted by motion features. Gao et al. [20] devised an adaptive attention model that makes the decoder adaptively select the visual information or the language context information to generate the visual words or the non-visual words in the caption. Long et al. [21] proposed a multi-faceted attention network to flexibly attend to the relevant frames, regions, and semantic attributes to generate the target words. The above attention-based methods all selected key regions or key frames only with the help of the language context. However, most of the words in a caption belong to non-visual words (e.g., Fig. 1). In this case, these methods cannot receive sufficient visual cues to focus on proper frames and thus failed to exactly align target words with corresponding frames. By contrast, we first introduce some visual tags to bridge the gap between vision and language. Then, a textual-temporal attention model is devised to precisely align the key frames with target words with the help of the language context plus the visual tags.

2.3. Exploiting visual tags

So far, the captioning methods exploiting visual tags can be categorized into two dimensions based on the roles they play in image or video captioning, that is, main and complementary roles.

Playing the main role You et al. [22] and Pan et al. [23] both exploited visual tags as main information in image and video captioning, respectively. In their methods, visual features were only used to initialize the LSTM. In terms of the use of visual tags, on one hand, You et al. [22] exploited attention mechanism to selectively focus on the proper visual tags to generate the target words. On the other hand, Pan et al. [23] devised a transfer unit to control the impacts of visual tags learned from images and videos.

Playing the complementary role Nian et al. [24] proposed to integrate visual tags into the visual feature to improve the S2VT model [12]. Aafaq et al. [25] proposed to use an object detector and a 3D-CNN to detect object and action tags to enrich the visual features. Hemalatha et al. [26] proposed to classify the videos into different domains and use the domain-specific semantic tags to enrich the visual features. Long et al. [21] proposed to introduce visual tags as the complementary information that aids in generating better captions. Yuan et al. [27] attempted to respectively utilize object, action, and global (i.e., object and action) tags to guide the appearance features, motion features, and generated words for target words prediction. Though the above methods all introduced visual tags to enhance captioning, all of them have not considered exploiting visual tags to bridge the gap between vision and language.

2.4. Summary

Our approach is built upon the attention-based CNN and RNN framework and aims for enhancing the alignment between target words and corresponding frames. However, different from those attention-based methods that merely rely on the language context to build the alignment, we first introduce some visual tags to bridge the gap between vision and language. Then, a textual-temporal attention model is devised to precisely align the key frames with target words with the help of the language context plus the visual tags. Furthermore, instead of using visual tags as the conventional main or complementary information for image or video captioning, we are the first work which makes them play a new role, i.e., bridge role, in video captioning.

3. Methodology

The overall framework of our proposed method is shown in Fig. 2. We first introduce the CNN encoder in Section 3.1. Then, we describe how to get visual tags in Section 3.2 in detail. Finally, we elaborate the proposed Textual-Temporal Attention Model (TTA) and how to incorporate it into the LSTM decoder in Section 3.3.

3.1. CNN encoder

In this paper, we utilize frame features extracted from a single frame, which represent static objects or scenes (e.g., “boy”, “cat”, and “sky”). Specifically, a source video is first uniformly sampled

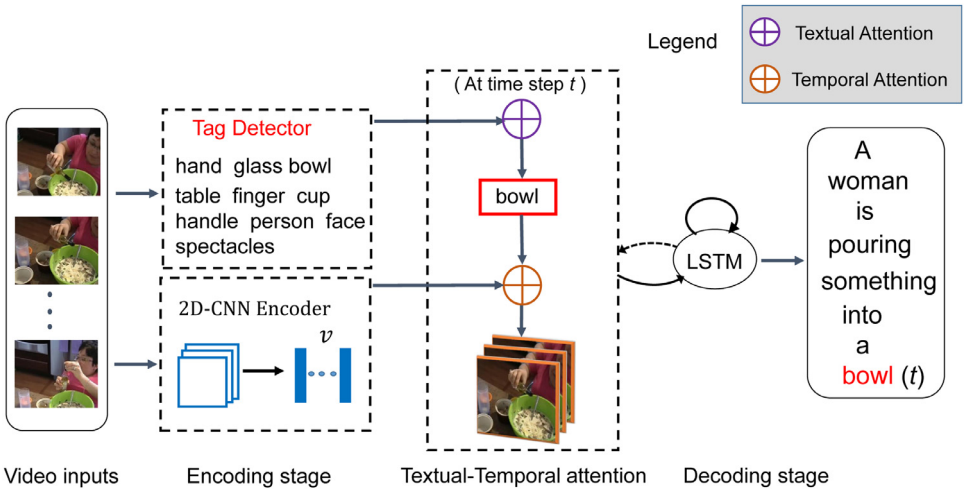

Fig. 2. The overall framework of our proposed Textual-Temporal Attention Model (TTA). Through TTA, at time step t, a subset of frames clearly containing the object "bowl" are determined with the guidance of the visual tag "bowl" plus the language context.

as a sequence of video frames $F = \{f_1, \ldots, f_k\}$. Each frame is fed into a pre-trained 2D CNN encoder and it outputs k frame features which are denoted as $v = \{v_1, v_2, \ldots, v_k\}$, in which $v_i \in \mathbb{R}^a$.

3.2. Visual tags

Visual tag vocabulary In our work, visual tags play an important role for bridging the gap between vision and language. In order to extract high-level visual tags from video frames, we need to first prepare a visual tag vocabulary to train the visual tag detector. Generally, a video captioning model generates a textual sentence by selecting words from the ground-truth captions on video captioning datasets. However, there are some objects in video frames but not in ground-truth captions. Therefore, we will construct the visual tag vocabulary from two sources, i.e., "in-domain" tags and "out-domain" tags. The "in-domain" tags are from the paired ground-truth captions on the MSR-VTT dataset [28] which is a large-scale video captioning dataset containing 200 K clip-sentence pairs. Specifically, we use the Stanford Parser to parse captions on the training set of MSR-VTT. Then, we choose the nsubj (nominal subject) and dobj (direct object) edges of the Stanford Parser outputs to find the subject-verb and verb-object pairs for each caption. Then, we extract subjects, verbs (ending with '-ing') and objects as candidate visual words. Please note that we manually remove all pronouns from these pairs, because they do not represent specific entities. Finally, we pick out 2500 most frequent visual words from those candidates as the "in-domain" tags.

Recently, Visual Genome (VG) dataset [29] has been widely used in the task of object detection, for it has 108,077 images, and over 7000 category classes annotated with object bounding boxes. The previous work of instance segmentation [30] has provided a list of 3000 most frequent classes on the VG dataset, and we find this list has covered all the 'in-domain' visual tags, which indicates VG contains the common visual tags on MSR-VTT. Besides, the remaining 500 classes are not in the ground truth captions of MSR-VTT. Hence, we will use all the classes contained in the list as our visual tag vocabulary which thus consists of 2500 "in-domain" visual tags and 500 "out-domain" visual tags.

Visual tag detector After constructing the visual tag vocabulary, we then need an effective visual tag detector to recognize candidate objects contained in each frame as many as possible. Recently, the region-based CNN (R-CNN) has made significant progress in detection task, especially Mask^X R-CNN [30] which aims at correctly detecting all objects in an image while outputting corresponding tags. Motived by this, we opt for the Mask^X R-CNN as our visual tag detector. The Mask^X R-CNN was trained based on the 3000 most frequent classes on VG. As our aforementioned, those classes consist of 2500 "in-domain" tags and 500 "out-domain" tags, so we follow their training procedure and use ResNet-101-FPN as the backbone network to train our visual tag detector. More details about the training are shown in Section 5 of [30] and its code page.[2]

Visual tag extraction Each visual tag is extracted based on its detected score in each frame and frequency of occurrence in all of the frames, respectively. To be more specific, given a sequence of video frames $F = \{f_1, \ldots, f_k\}$, we first feed them one by one into the pre-trained visual tag detector and it will output a set of object bounding boxes, object names and their corresponding detected scores in each frame. Next, we set 0.5 as a thresh score to pick out salient objects in each frame, where the score of each remained object is greater than or equal to this thresh score. After that, we will get a set of names of salient objects in each frame. Then, we rank each detected tag based on the number of appearance in all of the frames in descending order. Finally, the top-n most frequent visual tags are selected for each video. We denoted them as $tag = \{tag_1, tag_2, \ldots, tag_n\}$. If one of videos do not have n most frequent objects after above strategies, we pad the tag list with the unknown word sign, $< unk >$, to make enough n tags. Moreover, The n is a hyper-parameter and we will discuss it in Section 4.5.

Textual embedding In the task of video captioning, we aim to describe an input video with a textual sentence $S = \{w_1, w_2, \ldots, w_m\}$ which consists of m words. Let $\mathcal{W}_d$ denote the vocabulary including d words in the paired video-sentence data. Note that $\mathcal{W}_d$ has included three special token signs, $< sos >$, $< eos >$, and $< unk >$, to indicate "start of sentence", "end of sentence", and "unknown words", respectively. When generating the first word, we need a zeroth word w_0 (i.e., $< sos >$) to indicate "start of sentence". For the last word w_m, we denote it as $< eos >$. This is necessary because the model needs to know when to stop decoding during inference. For visual tag vocabulary, we denote the visual tag vocabulary as $\mathcal{W}_p$ including p (i.e., 3000 in our work) visual words. Hence, the whole vocabulary including g words is denoted as $\mathcal{W}_g = \mathcal{W}_d \cup \mathcal{W}_p$.

[2] https://github.com/ronghanghu/seg_every_thing.

Since both w_t and tag_j correspond to an entry in the vocabulary $\mathcal{W}_g$, they can be encoded with one-hot representations (1-of-g coding, g denotes the number of words in the vocabulary $\mathcal{W}_g$) in $\mathbb{R}^g$ space. To reduce parameter size, we project the one-hot representations into a low dimensional word vector space with a word embedding matrix E, where $E \in \mathbb{R}^{w \times g}$ and $w \ll g$. Finally, we can respectively represent tth word in the sentence S and jth visual tag in the video as a w-dimensional textual feature, i.e., $E[w_t] \in \mathbb{R}^w$ and $E[tag_j] \in \mathbb{R}^w$.

3.3. LSTM with textual-temporal attention model

Our decoder consists of the proposed textual-temporal attention model and a single LSTM. Moreover, a multilayer perceptron (MLP) layer is build upon the decoder to predict the probability distribution of target words.

LSTM decoder The structure of the LSTM is defined as:

$$\begin{aligned} i_t &= \sigma(W_i h_{t-1} + U_i x_t + b_i), \\ f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f), \\ o_t &= \sigma(W_o h_{t-1} + U_o x_t + b_o), \\ g_t &= tanh(W_f h_{t-1} + U_f x_t + b_g), \\ c_t &= c_{t-1} \odot f_t + i_t \odot g_t, \\ h_t &= o_t \odot tanh(c_t), \end{aligned} \tag{1}$$

where x_t is the input of the LSTM, h_t is the output of the LSTM, and h_{t-1} is the output of the LSTM at the $t-1$ time step. c_t is the memory state of the LSTM, and c_{t-1} is the previous memory state of the LSTM at the $t-1$ time step. σ is a *sigmoid* activation function and $\odot$ refers to element-wise multiplication. W_*, U_*, and b_* are the parameters to be learned. For simplicity, we refer to the operation procedure of an LSTM with the following notation:

$$h_t = LSTM(h_{t-1}, x_t). \tag{2}$$

In the initial time, the hidden state h_0 and memory state c_0 are computed by:

$$\begin{aligned} h_0, c_0 &= [W_{h0}, W_{c0}]\bar{V}, \\ \bar{V} &= \frac{1}{n}\sum_{i}^{k} v_i, \end{aligned} \tag{3}$$

where W_{h0} and W_{c0} are parameters to be learned.

In the decoding phase, each time the LSTM updates its hidden state h_t based on the previous hidden state h_{t-1} and word feature $E[w_{t-1}]$, as well as the currently attended visual tag $\varphi_t(E[tag])$ and the attended frame feature $\psi_t(v)$:

$$h_t = LSTM(h_{t-1}, E[w_{t-1}], \varphi_t(E[tag]), \psi_t(v)), \tag{4}$$

where the currently attended visual tag $\varphi_t(E[tag])$ and the attended frame feature $\psi_t(v)$ will be introduced as follows.

Textual-temporal attention model In the previous attention-based methods [4,5,20], the key frame feature at time step t is selected merely by the previous hidden state of the LSTM. However, most of the previously generated words are non-visual words. In this case, it is difficult to exactly align target words with relevant visual content.

To enhance their alignment, we first introduce visual tags to bridge the gap between these two modalities, because the tags extracted from a video not only belong to textual modality, but also are able to convey visual information. Next, a textual-temporal attention model (TTA) is devised to exactly align target words with corresponding frames, which is illustrated in Fig. 3. The proposed TTA consists of two stages: (1) the textual attention model is to select the most relevant visual tags according to the language context, and (2) the temporal attention model is to align the target words with the most relevant frames under the guidance of the selected visual tags plus the language context.

- Textual Attention. At each time step t, a key visual tag $\varphi_t(E[tag])$ is selected via relevant scores α which measure the relevance between the target word and every visual tags:

$$\varphi_t(E[tag]) = \sum_{j=1}^{n} \alpha_j^{(t)} E[tag_j]. \tag{5}$$

  Since visual tags and word features are homogeneous features, the relevance between them can be directly measured by the language context:

$$\begin{aligned} m_j^{(t)} &= U_a tanh(U_h h_{t-1} + U_s E[tag_j] + z), \\ \alpha_j^{(t)} &= softmax(m_j^{(t)}), \end{aligned} \tag{6}$$

  where U_a, U_h, U_s, and z are the parameters to be learned.
- Temporal Attention. In the second stage, as frame features and word features are heterogeneous features, we rely on the selected visual tags to bridge the gap between them. At each time step t, the relevance between frames and the target word is measured by $\varphi_t(E[tag])$ and h_{t-1}:

$$\begin{aligned} b_i^{(t)} &= W_b tanh(W_h h_{t-1} + W_a \varphi_t(E[tag]) + W_v v_i + b), \\ \beta_i^{(t)} &= softmax(b_i^{(t)}), \end{aligned} \tag{7}$$

  where W_b, W_h, W_a, W_v, and b are the parameters to be learned. Once the relevant scores $\beta_i^{(t)}$ are determined, each time the key frame is selected:

$$\psi_t(v) = \sum_{i=1}^{k} \beta_i^{(t)} v_i. \tag{8}$$

MLP layer The probability distribution of a series of target words at time step t will be obtained via a single hidden layer:

$$\begin{aligned} o &= W_e E[w_{t-1}] + W_y h_t + W_v \psi_t(v) + W_s \varphi_t(E[tag]), \\ \hat{w}_t &= softmax(U_y tanh(o) + b_y), \end{aligned} \tag{9}$$

where W_*, U_y, and b_y are the parameters to be learned.

4. Experiments

4.1. Datasets

The microsoft video description corpus (MSVD) MSVD [31] has 1970 video clips. Each video clip is provided with about 41 human annotated sentences. Following [4], the dataset can be divided into a training set of 1200 video clips, a validation set of 100 clips, and a test set consisting of the rest of 670 clips.

MSR video to text (MSR-VTT) MSR-VTT [28] consists of 10,000 video clips from 20 general categories. Each video clip is provided with 20 human annotated sentences. There are 200 K clip-sentence pairs corresponding to 1.8 M words. We use the official spilt with 6513 videos for training, 497 for validation and 2990 for testing.

4.2. Evaluation metrics

We use four standard metrics to evaluate the quality of generated sentences, i.e., BLEU [32], METEOR [33], ROUGE-L [34] and CIDEr [35].

BLEU analyzes the co-occurrences of n-grams between the candidate and reference sentences. METEOR can generate an alignment according to exact token matching to judge the word correlation between candidate and reference sentences. ROUGE-L uses a measure based on the Longest Common Subsequence (LCS), which is a set words shared by two sentences which occur in the same order. CIDEr exploits human consensus to evaluate video descriptions. We get all the results in this paper according to the Microsoft COCO evaluation server [36].

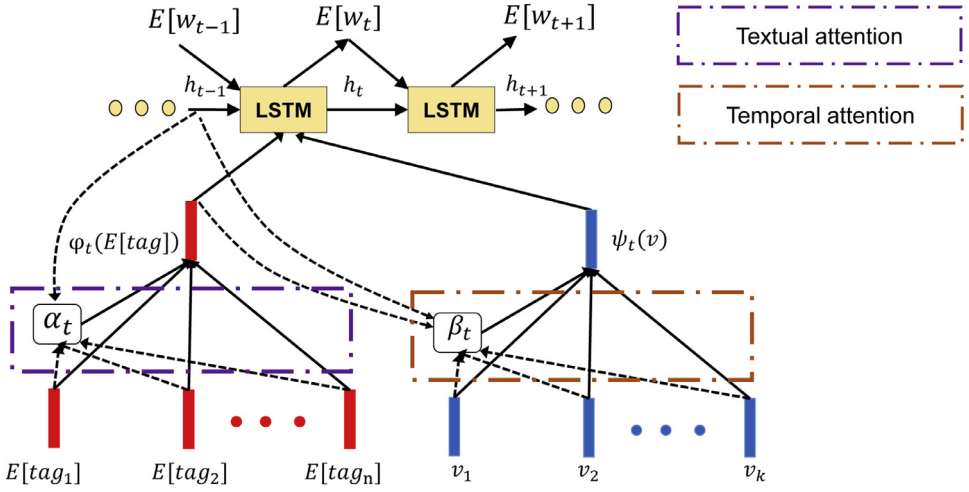

Fig. 3. Illustration of the proposed textual-temporal attention mechanism (TTA).

4.3. Implementation details

The whole vocabulary size consists of the words of visual tags and the words on the video captioning dataset, as well as three special word signs, $< sos >$, $< eos >$, and $< unk >$. Thus, after preprocessing, the whole vocabulary size is 14,021 for MSVD and 29,552 for MSR-VTT, respectively.

The dimension of the word embedding is set to 468 for both datasets, so tth word in the sentence S and jth visual tag in the video can be represented as a 468-dimensional textual feature, respectively. The hidden layer size of the LSTM is all set to 512. The number of visual tags is set to 10. For feature extraction, we first select 28 equally-spaced frames for each video. Then, we extract frame features from two kinds of CNN encoder, that is, VGG [37] and ResNet-152 [38] both pre-trained on the Imagenet dataset [39], where the dimension of frame features is 4096 and 2048, respectively.

Model training In the training phase, we set the mini-batch size as 64 on MSVD and 128 on MSR-VTT. The learning rate is 2e−5 on MSVD and 2e−4 on MSR-VTT, respectively. Moreover, we set dropout regularization in the rate of 0.5 in all layers and cip graients element wise at 5. The maximum training iteration is set as 60 epochs on the both datasets. We apply Adadelta algorithm [40] to minimize the negative log-likelihood loss:

$$L(\theta) = -\sum_{t=1}^{m} logp(w_t|w_{<t}, v, tag, \theta), \tag{10}$$

where θ are parameters of the video captioning model and m is the length of a sentence. When finishing the training phase, the minimum loss value on the validation set is used as a metric to choose the best model for testing.

Inference During inference, the ground truths are not provided, we would actually need to input the start sign word $< sos >$ to start decoding and feed the previously generated word to the LSTM at each time step until the end sign word $< eos >$ is reached. At each time step, the straightforward option would be to choose the word with the maximum score after *softmax* and use it to predict the next word. But this is not optimal because the rest of the sequence hinges on that first word. If that choice isn't the best, everything that follows is sub-optimal. Therefore, we use beam search with size 5 to choose the sequence that has the highest overall score in 5 candidate sequences.

Table 1
Ablation study results obtained on MSVD, where B-4, M, R-L, and C are short for BLEU-4, METEOR, ROUGE-L, and CIDEr. All values are reported as percentage (%). R152 denotes ResNet-152.

	MSVD			
Method	B-4	M	R-L	C
TA (R152)	47.7	33.4	70.1	80.8
TA+tag$_{mean}$ (R152)	**50.8**	33.6	69.6	80.3
(relative improvement%↑)*	**(6.5%↑)**	(0.6%↑)	(-0.71%↑)	(-6.2%↑)
TTA (R152)	50.6	**33.8**	**70.4**	**83.0**
(relative improvement%↑)	(6.1%↑)	**(1.2%↑)**	**(0.43%↑)**	**(2.7%↑)**

*TA+tag$_{mean}$, and TTA have relatively improvements (%↑) over the basis TA.

4.4. Ablation studies

In order to figure out the contribution of each component, we conduct the following ablation studies on the both datasets. All the studies take ResNet-152 as the encoder.

- Basis (yao et al. [4]). We use [4] as our basis. In their method, they exploited the temporal attention mechanism to select the key frames to generate target words, where the relevance between two modalities are merely measured by the language context.
- Basis + visual tags (mean-pooling). To verify the effectiveness of visual tags, we introduce visual tags to bridge the gap between vision and language. In this study, we perform mean-pooling strategy over all visual tags and only use the temporal attention like Basis.
- Basis + visual tags (textual-temporal attention). In this study, according to the proposed textual-temporal attention model, we first select the key visual tags via the language context, and then select the key frames via the key visual tags plus the language context.

For convenience, we denote three studies as: TA, TA+tag$_{mean}$, TTA, respectively. The experimental results are shown in Table 1 and Table 2.

The influence of visual tags From Table 1, on MSVD, we can see that, although TA+tag$_{mean}$ has a little descent on ROUGE-L and CIDEr, it achieves better scores on other metrics than TA, in particular with an increase of 6.5% on BLEU-4. From Table 2, on MSR-VTT, we can see that TA+tag$_{mean}$ outperforms TA on all of the metrics. Experimental results validate our claim, that is, visual tags are effectively capable of bridging the gap between vision and lan-

Table 2
Ablation study results obtained on MSR-VTT, where B-4, M, R-L, and C are short for BLEU-4, METEOR, ROUGE-L, and CIDEr. All values are reported as percentage (%).

	MSR-VTT			
Method	B-4	M	R-L	C
TA (R152)	38.6	26.3	58.6	42.0
TA+tag$_{mean}$ (R152)	39.4	26.5	59.8	43.1
(relative improvement%↑)*	(2.1%↑)	(0.76%↑)	(2.0%↑)	(2.6%↑)
TTA (R152)	**39.6**	**27.0**	**60.2**	**45.7**
(relative improvement%↑)	**(2.6%↑)**	**(2.7%↑)**	**(2.7%↑)**	**(8.8%↑)**

*TA+tag$_{mean}$, and TTA have relatively improvements (%↑) over the basis TA.

Table 3
Evaluation on different dimensions and initializing methods for word vectors on MSVD.

Model	Initializing	Word Dimension	B-4	M	R	C
TTA (R152)	random	256	51.3	33.8	**70.6**	82.3
TTA (R152)	random	300	**52.0**	**34.0**	70.5	81.2
TTA (R152)	random	468	50.6	33.8	70.4	83.0
TTA (R152)	random	512	50.1	33.6	69.6	82.1
TTA (R152)	Glove	300	50.6	33.2	69.7	**84.1**
TTA (R152)	Word2Vec	300	51.8	**34.0**	69.9	82.6

guage and thus help the temporal attention model to enhance the alignment between target words and relevant frames.

The influence of textual-temporal attention From Table 1, we can see that TA+tag$_{mean}$ got lower scores than TA on ROUGE-L and CIDEr. Our conjecture is that it is useful to bridge the gap between target words and relevant frames by introducing visual tags. However, as visual tags belong to fine-grained information, simply averaging them may weaken their discrimination. From both Tables 1 and 2, it is clearly observe that TTA makes a significant improvements on all of the metrics. This verifies the effectiveness of proposed textual-temporal attention model. Through this model, the proper visual tag is able to help the temporal attention model to exactly align each target word with its relevant frame as far as possible.

4.5. Evaluation on different hyper-parameters

In this section, we perform three comparative experiments to discuss sensitivity to hyper-parameters, i.e., (1) the dimension of word vector and initializing with pre-trained word vector; (2) the size of beam search; (3) the number of visual tags.

Evaluation on word vector How many dimensions and initializing with what kind of word vector are both important for video caption generation. To this end, we first analyze the effect of different dimensions, i.e. 256, 300, 468, and 512, for randomly initializing the word vectors. And then rather than randomly initializing, we will utilize two kinds of pre-trained word vectors, Glove [41] and Word2Vec [42], for initializing the word vectors. The Glove model was pre-trained on a corpus of 840 billion words and contains a vocabulary of 2.2 million words, where each word is represented by a 300-dimensional vector. For Word2Vec, it was pre-trained on a part of Google News corpus (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

Tables 3 and 4 show the comparative results with different word dimensions and initializing methods for word vectors on MSVD and MSR-VTT, respectively. On MSVD, we observe that it achieves the best performances when setting the dimension of word embedding to 300 and leveraging the randomly initializing method. On MSR-VTT, on the contrary, the proper dimension of word embedding is 468. According to these experiments, we respectively set the dimension of word embedding as 300 on MSVD and 468 on MSR-VTT to conduct our following experiments.

Table 4
Evaluation on different dimensions and initializing methods for word vectors on MSR-VTT.

Model	Initializing	Word Dimention	B-4	M	R	C
TTA (R152)	random	256	39.1	26.0	59.1	43.0
TTA (R152)	random	300	38.8	25.8	58.3	43.4
TTA (R152)	random	468	39.6	**27.0**	**60.2**	**45.7**
TTA (R152)	random	512	38.0	26.6	59.3	43.4
TTA (R152)	Glove	300	39.7	26.6	59.2	44.5
TTA (R152)	Word2Vec	300	**40.2**	26.5	59.3	43.1

Evaluation on the size of beam search The beam search algorithm is used during the inference time, and different beam sizes also have different impacts for finally generating captions. In order to obtain the proper beam sizes, we set the beam size as 3, 4, 5, 6, and 7 to validate their effects. Table 5 shows the comparative results on the both datasets. We observe that 6 and 5 are the best size on MSVD and MSR-VTT, respectively. Thus, we finally choose that the beam size is equal to 6 and 5 on MSVD and MSR-VTT respectively to conduct the following experiments.

Evaluation on the number of visual tags How many visual tags to select is also critical for generating captions. To obtain proper number of visual tags, we select 5, 10, 15, and 20 visual tags of each video to verify the performance, respectively. Fig. 4 illustrates the results of MSVD and MSR-VTT with different top-n visual tags. Each sub-figure shows the scores of different n with the same metric and each color shows the scores of the same n with different metrics.

Frome Fig. 4, we can observe that the best performances are achieved on the both datasets when selecting top-10 visual tags in each video. The reason is that when the number of tags is smaller, some key objects might be missing. When the number is larger, on the contrary, some trivial visual tags might be noises when using them to bridge the gap between vision and language. Hence, we finally select top-10 visual tags in each video on the both datasets to conduct the following experiments.

4.6. Comparison with the methods exploring the alignment between vision and language

In Section 2.2, we have qualitatively compared our proposed methods with the methods exploring the alignment between vision and language. Those methods are classified into two dimensions. On one hand, Pan et al. (LSTM-E) [16] and Guo et al. (aLSTMs) [17] both belong to embedding-based alignment which explores the alignment between the whole sentence and the whole video. On the other hand, Tu et al. (STAT) [18], Hori (AF) et al. [19] Long et al. [21] and Gao et al. [20] build the alignment between target words and corresponding frames based on the attention mechanism. In this subsection, we will quantitatively compare TTA with them. Since LSTM-E and aLSTMs did not report the results on the MSR-VTT dataset, we only conduct comparison experiments on the MSVD dataset. Note that for fair comparison, we additionally used VGG as the encoder for TTA, which used default hyper-parameters elaborated in Section 4.3.

The comparison results are reported in Table 6. By comparing TTA with LSTM-E and aLSTMs, we can see that TTA outperforms them on all of the metrics. Unlike their alignment building upon the whole sentence and the whole video, our alignment is built between word-level and frame-level. This further shows that we should build the alignment between vision and language as exact as possible.

Moreover, STAT, AF, MFATT, and hLSTMat all merely exploited the language context to determine key visual or semantic information. Compared to STAT and AF, TTA (VGG19) achieved marked

Table 5
Evaluation on the size of beam search on MSVD and MSR-VTT.

		MSVD				MSR-VTT			
Model	Size	B-4	M	R	C	B-4	M	R	C
TTA(R152)	3	46.0	33.2	69.4	78.5	**39.8**	26.9	59.9	43.8
TTA(R152)	4	50.5	34.1	69.8	80.2	39.4	**27.0**	59.8	44.8
TTA(R152)	5	**52.0**	34.0	**70.5**	81.2	39.6	**27.0**	**60.2**	**45.7**
TTA(R152)	6	51.7	**34.5**	70.2	**81.8**	37.4	25.8	58.4	43.2
TTA(R152)	7	51.0	**34.5**	70.3	80.6	39.5	25.6	58.7	43.6

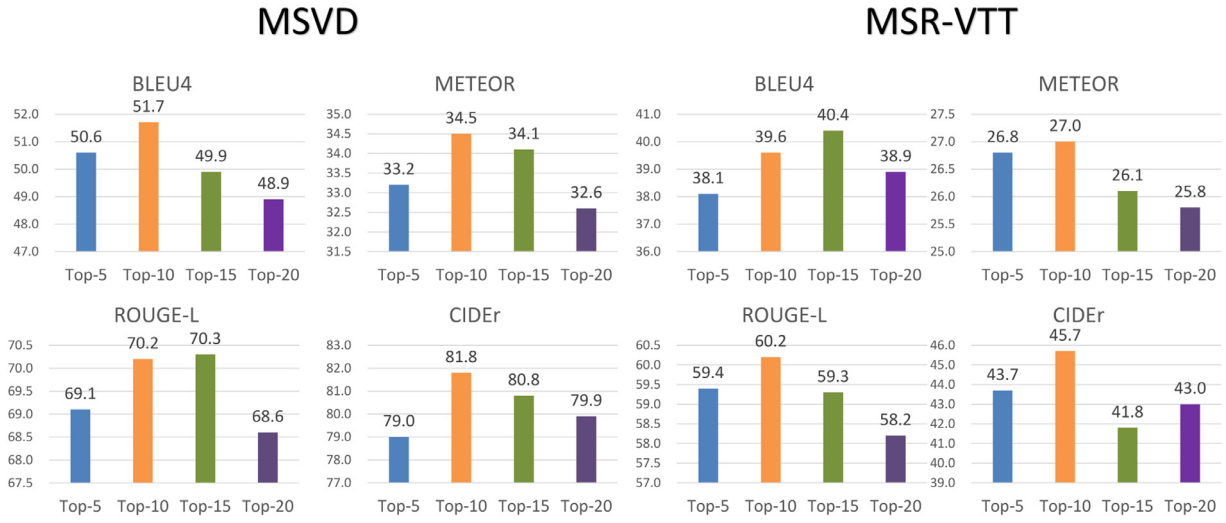

Fig. 4. Evaluation on the number of visual tags on MSVD and MSR-VTT.

Table 6
Performances of proposed method and the methods exploring the alignment between vision and language on MSVD, where V16, V19, G, and C denote VGG-16, VGG-19, GoogleNet, and C3D. The symbol '–" indicates such metric is unreported.

Method	B-4	M	R-L	C
LSTM-E (V19+C) [16]	45.3	31.0	–	–
aLSTMs (G+C) [17]	44.9	30.4	–	60.1
STAT (G+C) [18]	51.1	32.7	–	67.5
AF (V16+C) [19]	52.4	32.0	–	68.8
MFATT [21] (R152+C)	52.0	33.5	–	72.1
hLSTMat [20] (R152+C)	**54.0**	33.2	–	71.3
TTA (V16)	46.2	32.4	67.9	70.2
TTA (V19)	48.4	32.5	69.4	74.9
TTA(R152)	51.7	**34.5**	**70.2**	**81.8**

Table 7
Comparison with the State-of-the-art Methods on the MSVD dataset, where V16, R-152, IRV2, C, IV4, and I denote VGG-16, ResNet-152, Inception-ResNet-V2, C3D, InceptionV4 and I3D, respectively. The symbol "–" indicates such metric is unreported.

Method	B-4	M	R-L	C
Att-TVT [14] (R152+I) (2018)	53.0	34.7	71.7	80.8
MFATT [21] (R152+C) (2018)	52.0	33.5	–	72.1
RecNet$_{local}$ [43] (IV4) (2018)	52.3	34.1	69.8	80.3
LG-DenseLSTM [45](V16+C) (2019)	50.4	32.9	69.9	72.6
GRU-EVE$_{hft+sem}$ [25] (IRV2+C) (2019)	47.9	35.0	71.5	78.1
TDConvED [44] (R152) (2019)	53.3	33.8	–	76.4
SAAT [46] (IRV2+C) (2020)	46.5	33.5	69.4	81.0
hLSTMat [20] (R152+C) (2020)	**54.0**	33.2	–	71.3
TTA (R152)	51.7	34.5	70.2	81.8
TTA (R152+C)	51.8	**35.5**	**72.4**	**87.7**

improvements on CIDEr. In terms of METEOR, TTA (VGG19) only a little lower than STAT but better than AF. However, both of them use two kinds of features (VGG / GoogleNet + C3D), while we only use VGG features. Compared to MFATT and hLSTMat, our TTA (R152) significant outperforms them on METEOR and CIDEr. Note that MFATT also used visual tags but used them as complementary information. All in all, experimental results demonstrate the effectiveness of our proposed methods.

4.7. Comparison with the state-of-the-art methods

In this subsection, we compare our proposed methods with recently published state-of-the-art methods:

1) Att-TVT [14] which proposed to use Transformer for video captioning.
2) MFATT [21] which jointly leveraged multiple sorts of visual features and semantic attributes.
3) RecNet$_{local}$ [43] which proposed a local reconstructor architecture to use the back flow between the encoder and the decoder.
4) GRU-EVE$_{hft+sem}$ [25] which embedded rich temporal dynamics in visual features by hierarchically applying Short Fourier Transform to CNN features of the whole video.
5) TDConvED [44] which aimed to fully employ convolutions in both encoder and decoder networks.
6) LG-DenseLSTM [45] which proposed a dense LSTM for video captioning.
7) hLSTMat [20] which proposed a hierarchical LSTM with adaptive attention.
8) SAAT [46] which proposed a syntax-aware action targeting (SAAT) module.

Results on the MSVD dataset We report the results in Table 7. For fair comparison, we also additionally use C3D [47] to extract motion features and then concatenate them with frame features to input the proposed TTA model. We first compare our method with

Table 8
Comparison with the State-of-the-art Methods on the MSR-VTT dataset, where V16, R152, IRV2, C, IV4, I, N, and FR denote VGG-16, ResNet-152, Inception-ResNet-V2, C3D, IncepctionV4, I3D, NasNet and Faster R-CNN, respectively.The symbol "–" indicates such metric is unreported.

Method	B-4	M	R-L	C
Att-TVT [14] (N+I) (2018)	40.1	27.9	59.6	47.7
MFATT [21] (R152+C) (2018)	39.1	26.7	–	–
RecNet$_{local}$ [43] (IV4) (2018)	39.1	26.6	59.3	42.7
LG-DenseLSTM [45](V16+C) (2019)	37.6	26.2	–	42.0
GRU-EVE$_{hft+sem}$ [25] (IRV2+C) (2019)	38.3	**28.4**	60.7	48.1
TDConvED [44] (R152) (2019)	39.5	27.5	–	42.8
SAAT [46] (IRV2+C+FR) (2020)	40.5	28.2	60.9	**49.1**
hLSTMat [20] (R152+C) (2020)	38.7	26.8	–	41.9
TTA (R152)	39.6	27.0	60.2	45.7
TTA (R152+C)	**41.4**	27.7	**61.1**	46.7

the methods based on the recently proposed transformer framework. Furthermore, we compare our method with the methods based on the recurrent neural networks. The comparison results are as follows:

1) Compared to Att-TVT (R152+I3D), except BLEU-4, our TTA (R152+C3D) outperforms it on the other metrics. Especially on CIDEr, TTA has over 8.5% improvement.

2) Compared to RecNet$_{local}$ (IV4) and TDConvED (R152) which used a single CNN encoder, except BLEU-4, TTA (R152) outperforms them on the other metrics. Compared to LG-DenseLSTM (V16+C) using a shallow CNN with C3D, our TTA (R152) also outperforms them on all of the metrics. Compared to others using deep CNNs with C3D, our TTA (R152+C3D) also achieves best performances on METEOR, ROUGE-L and CIDEr, which indicate the effectiveness of our proposed method.

Results on the MSR-VTT dataset We report the results in Table 8. Like MSVD, we also compare our methods with the state-of-the-art methods in two dimensions, that is, using the transformer framework and the recurrent framework. The comparison results are as follows:

1) Compared to Att-TVT (N+I3D), TTA (R152+C3D) is only better than it on BLEU-4 and ROUGE-L. One of possible reasons is that Att-TVT selected NasNet on this dataset and they explained in the paper that NasNet achieves a higher accuracy on the image classification problem.

2) Compared to these methods using the encoder-decoder framework, TTA outperforms RecNet$_{local}$, LG-DenseLSTM, hLSTMat and MFATT on all the metrics. Compared to TDConvED (R152), except METEOR, TTA (R152) outperforms it on the other metrics. Especially on CIDEr, TTA has 6.8% improvement. Moreover, we observe that TTA both outperforms GRU-EVE$_{hft+sem}$ and SAAT on BLEU-4 and ROUGE-L. Above all, compared to these state-of-the-art methods, our TTA also achieves superior performances.

4.8. Qualitative analysis

In order to better understand our proposed TTA, Fig. 5 shows 5 examples with detected top-10 visual tags, human-annotated ground truth captions, and captions generated by two methods, i.e., TA and TTA. TA only builds the coarse alignment between target words and corresponding frames via a temporal attention model. TTA enhances the alignment between target words and corresponding frames via visual tags and the proposed textual-temporal attention model. From Fig. 5, we can intuitively see that although TA can generate logically correct captions to describe videos, some objects are missing or wrongly recognized. Instead, TTA can give the decoder more visual cues to focus on the key frames containing these objects (e.g., "table", "bird", and "face"), so it is able to generate captions more accurate than TA. For example, in the first video, if only relying on the generated non-visual word "a", it is difficult for the decoder to focus on the frames clearly containing "girl". In this case, she is wrongly recognized as a boy in the generated caption by TA. However, if we first give the decoder the visual cue "girl", it is able to easily attend to the corresponding frames. Similarly, in the generated captions for the second to fourth videos by TA, the key objects "bird", "table", and "face" are missing. By comparison, the generated captions by TTA successfully captures these objects. Moreover, in the last video, TTA can detailedly depict " a man in a blue shirt" with the help of the detected tags "man" and "shirt". Compared to TA, the alignment of TTA is more exact and thus it can generate more comprehensive captions.

In Fig. 6, we further visualize the two kinds of attention weights shift about a test video in Fig. 5. One is the temporal attention shift

Fig. 5. Five examples from the test sets of MSVD and MSR-VTT, which also involve detected top-10 visual tags in the videos, human-annotated ground truth captions (GT) and the captions generated by two methods, i.e., temporal attention (TA) and textual-temporal attention (TTA).

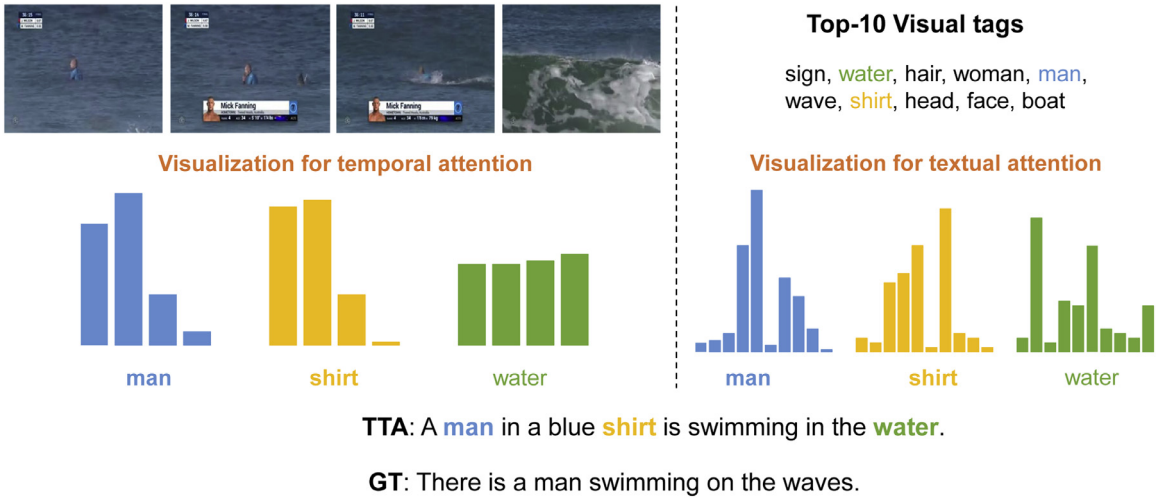

Fig. 6. Visualization of sampled frames, generated and ground truth captions, visual tags and corresponding attentions shift about a test video on the MSR-VTT dataset.

on the sampled frames with respect to each key word in the generated caption. The other is the textual attention shift on the top-10 visual tags with regard to each key word in the generated caption. From Fig. 6, we can intuitively see that the both attentions shift of each key word in the generated caption are much consistent to the video content and the visual tags. For example, when generating the word "man", the temporal attention model mainly focuses on the second frame, and the textual attention model will give the visual tag "man" more attention weights.

5. Conclusion and discussion

In this paper, we propose to exploit visual tags to bridge the gap between vision and language for video captioning. In order to exactly align the target words with corresponding frames, we present a textual-temporal attention model (TTA), where the language context is first utilized to select the key visual tags, and then the key frames are selected with the guidance of these key visual tags plus the language context. Extensive comparison experiments are conducted on MSVD and MSR-VTT. Experimental results show that our method achieves the state-of-the-art performances.

Besides, there are also other directions which are worth further exploring based on this work in the future. For instance, one is to introduce visual tags and incorporate the proposed TTA into the newly proposed transformer framework which has achieved state-of-the-art performance in many multi-modal tasks. Another is to study and analyze the diversity of the generated captions as done in Chen et al. [48], for repeated or similar captions are often generated in the most existing methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work was supported by National Natural Science Foundation of China (Grant nos. 61732005, 61672271, 61761026, 61866020, 61972186, and 61762056), National Key Research and Development Plan (Grant nos. 2019QY1801, 2019QY1802, and 2019QY1800), Yunnan high-tech industry development project (Grant no. 201606), Natural Science Foundation of Yunnan Province (Grant no. 2018FB104), and Talent Fund for Kunming University of Science and Technology (Grant no. KKSY201703005).

References

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[2] X. Bai, B. Shi, C. Zhang, X. Cai, L. Qi, Text/non-text image classification in the wild with convolutional neural networks, Pattern Recognit. 66 (2017) 437–446.
[3] I. Sutskever, O. Vinyals, Q. Le, Sequence to sequence learning with neural networks, Adv. NIPS (2014) 3104–3112.
[4] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: ICCV, 2015, pp. 4507–4515.
[5] X. Zhang, K. Gao, Y. Zhang, D. Zhang, J. Li, Q. Tian, Task-driven dynamic fusion: reducing ambiguity in video description, in: CVPR, 2017, pp. 3713–3721.
[6] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, Q. Dai, STAT: spatial-temporal attention mechanism for video captioning, IEEE Trans. Multimed. 22 (1) (2020) 229–241.
[7] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473(2014).
[8] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, B. Schiele, Translating video content to natural language descriptions, in: ICCV, 2013, pp. 433–440.
[9] R. Xu, C. Xiong, W. Chen, J.J. Corso, Jointly modeling deep video and compositional text to bridge vision and language in a unified framework, in: AAAI, 2015, pp. 2346–2352.
[10] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, K. Saenko, Translating videos to natural language using deep recurrent neural networks, arXiv preprint arXiv:1412.4729(2014).
[11] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[12] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, Sequence to sequence-video to text, in: ICCV, 2015, pp. 4534–4542.
[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.
[14] M. Chen, Y. Li, Z. Zhang, S. Huang, Tvt: two-view transformer network for video captioning, in: ACML, 2018, pp. 847–862.
[15] L. Zhou, Y. Zhou, J.J. Corso, R. Socher, C. Xiong, End-to-end dense video captioning with masked transformer, in: CVPR, 2018, pp. 8739–8748.
[16] Y. Pan, T. Mei, T. Yao, H. Li, Y. Rui, Jointly modeling embedding and translation to bridge video and language, in: CVPR, 2016, pp. 4594–4602.
[17] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, H.T. Shen, Attention-based LSTM with semantic consistency for videos captioning, in: ACM MM, 2016, pp. 357–361.
[18] Y. Tu, X. Zhang, B. Liu, C. Yan, Video description with spatial-temporal attention, in: ACM MM, 2017, pp. 1014–1022.
[19] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J.R. Hershey, T.K. Marks, K. Sumi, Attention-based multimodal fusion for video description, in: ICCV, 2017, pp. 4193–4202.
[20] L. Gao, X. Li, J. Song, H.T. Shen, Hierarchical LSTMs with adaptive attention for visual captioning, IEEE Trans. Pattern Anal. Mach. Intell. 42 (5) (2020) 1112–1131.
[21] X. Long, C. Gan, G. de Melo, Video captioning with multi-faceted attention, Trans. Assoc. Comput. Linguist. 6 (2018) 173–184.
[22] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: CVPR, 2016, pp. 4651–4659.
[23] Y. Pan, T. Yao, H. Li, T. Mei, Video captioning with transferred semantic attributes, in: CVPR, 2017, pp. 6504–6512.

[24] F. Nian, T. Li, Y. Wang, X. Wu, B. Ni, C. Xu, Learning explicit video attributes from mid-level representation for video captioning, Comput. Vis. Image Underst. 163 (2017) 126–138.
[25] N. Aafaq, N. Akhtar, W. Liu, S.Z. Gilani, A. Mian, Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning, in: CVPR, 2019, pp. 12487–12496.
[26] M. Hemalatha, C.C. Sekhar, Domain-specific semantics guided approach to video captioning, in: WACV, 2020, pp. 1576–1585.
[27] J. Yuan, C. Tian, X. Zhang, Y. Ding, W. Wei, Video captioning with semantic guiding, in: BigMM, 2018, pp. 1–5.
[28] J. Xu, T. Mei, T. Yao, Y. Rui, Msr-vtt: a large video description dataset for bridging video and language, in: CVPR, 2016, pp. 5288–5296.
[29] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, et al., Visual genome: connecting language and vision using crowdsourced dense image annotations, in: IJCV, 2017, pp. 32–73.
[30] R. Hu, P. Dollár, K. He, T. Darrell, R. Girshick, Learning to segment every thing, in: CVPR, 2018, pp. 4233–4241.
[31] D.L. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: ACL, 2011, pp. 190–200.
[32] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: ACL, 2002, pp. 311–318.
[33] S. Banerjee, A. Lavie, Meteor: an automatic metric for MT evaluation with improved correlation with human judgments, in: ACL, 2005, pp. 65–72.
[34] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.
[35] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, in: CVPR, 2015, pp. 4566–4575.
[36] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: data collection and evaluation server, arXiv preprint arXiv:1504.00325(2015).
[37] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556(2014).
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, IJCV (2015) 211–252.
[40] M.D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701(2012).
[41] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
[42] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781(2013).
[43] B. Wang, L. Ma, W. Zhang, W. Liu, Reconstruction network for video captioning, in: CVPR, 2018, pp. 7622–7631.
[44] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, T. Mei, Temporal deformable convolutional encoder-decoder networks for video captioning, in: AAAI, 2019, pp. 8167–8174.
[45] Y. Zhu, S. Jiang, Attention-based densely connected LSTM for video captioning, in: ACM MM, 2019, pp. 802–810.
[46] Q. Zheng, C. Wang, D. Tao, Syntax-aware action targeting for video captioning, in: CVPR, 2020, pp. 13096–13105.
[47] K. Hara, H. Kataoka, Y. Satoh, Can spatiotemporal 3D CNNs retrace the history of 2DCNNs and imagenet? in: CVPR, 2018, pp. 6546–6555.
[48] H. Chen, J. Li, X. Hu, Delving deeper into the decoder for video captioning, arXiv preprint arXiv:2001.05614(2020).

Yunbin Tu Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, Kunming 650500, P.R. China E-mail: tuyunbin1995@foxmail.com He received the bachelor's degree in Automation from Hangzhou Dianzi University. He is currently pursuing a M.S degrees in Pattern Recognition and Intelligent System at Kunming University of Science and Technology. His research interests include deep learning and video captioning.

Chang Zhou Department of information science and technology, Tsinghua Shenzhen International Graduate School, Guangdong, Shenzhen 518000, P.R. China E-mail: zhouc18@mails.tsinghua.edu.cn Chang Zhou: He received the bachelor's degree in Electrical engineering and its automation from China University of Mining and Technology. He is studying for a M.S degrees in control engineering from Tsinghua University. His research interests include image processing and computer vision.

Junjun Guo Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, Kunming 650500, P.R. China E-mail:guojjgb@163.com Junjun Guo: He graduated with Ph.D. degree in engineering from Xi'an Jiaotong University. He is working as a lecturer of Kunming University of Technology. His main research interest is Multi-model information fusion.

Shengxiang Gao Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, Kunming 650500, P.R. China E-mail:gaoshengxiang.yn@foxmail.com Shengxiang Gao: She received the M.S. degree in Pattern Recognition and Intelligent System and the Ph.D. degree in Control Engineering from Kunming University of Science and Technology in 2005 and 2016, respectively. She is currently associate professor in School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. Her main research interests include machine learning, nature language processing and machine translation.

Zhengtao Yu Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, Kunming 650500, P.R. China E-mail:ztyu@hotmail.com Zhengtao Yu: He received his Ph.D. degree in computer application technology from Beijing Institute of Technology, Beijing, China, in2005. He is currently a professor in the School of Information Engineering and Automation, Kunming University of Science and Technology, China. His main research interests include natural language processing, information retrieval and machine learning.