



# Double-channel language feature mining based model for video description

Pengjie Tang<sup>1,2</sup> · Jiewu Xia<sup>1,2</sup> · Yunlan Tan<sup>1,2</sup> · Bin Tan<sup>1,2</sup>

Received: 2 January 2020 / Revised: 15 July 2020 / Accepted: 20 August 2020 /  
Published online: 31 August 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Video description is to translate video to natural language. Many recent effective models for the task are developed with the popular deep convolutional neural networks and recurrent neural networks. However, the abstractness and representation ability of visual motion feature and language feature are usually ignored in most of popular methods. In this work, a framework based on double-channel language feature mining is proposed, where deep transformation layer (DTL) is employed in both of the stages of motion feature extraction and language modeling, to increase the number of feature transformation and enhance the power of representation and generalization of the features. In addition, the early deep sequential fusion strategy is introduced into the model with element-wise product for feature fusing. Moreover, for more comprehensive information, the late deep sequential fusion strategy is also employed, and the output probabilities from the modules with DTL and without DTL are fused with weight average for further improving accuracy and semantics of generated sentence. Multiple experiments and ablation study are conducted on two public datasets including Youtube2Text and MSR-VTT2016, and competitive results compared to the other popular methods are achieved. Especially on CIDEr metric, the performance reaches to 82.5 and 45.9 on the two datasets respectively, demonstrating the effectiveness of the proposed model.

**Keywords** Double-channel · Language feature · Video description · LSTM · Deep fusion

---

Research Foundation of Art Planning of Jiangxi Province (No. YG2017283); Bidding Project for the Foundation of Colleges Key Research on Humanities and Social Science of Jiangxi Province (No. JD17082); The Doctoral Scientific Research Foundation of Jinggangshang University (No. JZB1923, JZB1807); National Natural Science Foundation of P. R. China (No. 61762052).

✉ Jiewu Xia  
jwxia@126.com

Extended author information available on the last page of the article.

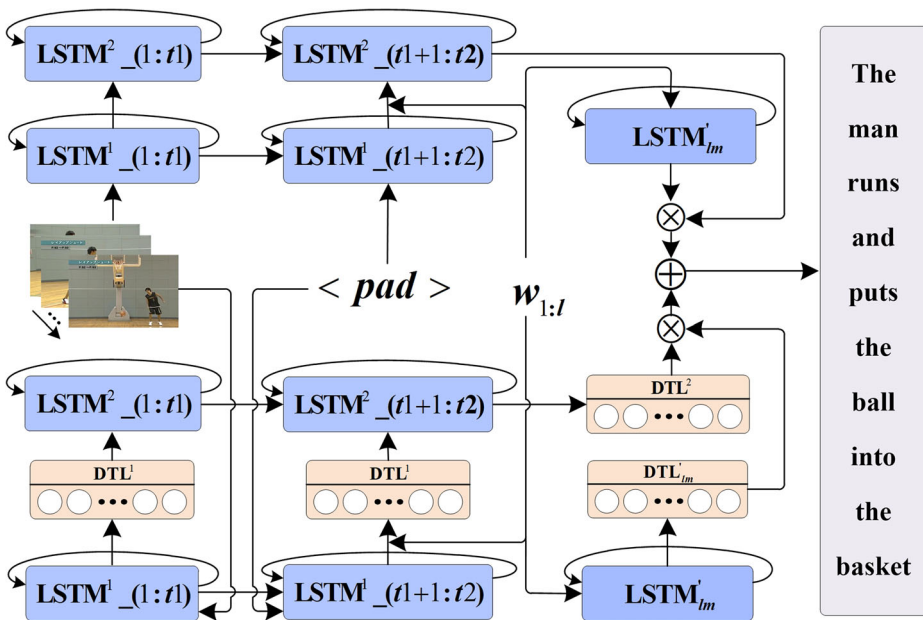
# 1 Introduction

Video description aims to translate and re-express the visual content with natural language, which belongs to a high-level understanding task in computer vision since the representational visual data is transformed into more abstract language. And it has bright prospect in early education, visual assistant, automatic explanation and intelligent interactive environment development. However, the task depends on various techniques of computer vision, as well as methods in the field of natural language processing, resulting in complicated process and more challenges.

So far there are diversified frameworks and models for bridging the vision to language. In the early days, the template based frameworks [17, 23] and semantic transferring based frameworks [51] are developed to generate sentence with stereotyped structure or similar samples from query database built in advance. Actually, the descriptions usually lack flexibility in sentence pattern and using word, leading to insufficient semantics. Afterwards, inspired by the task of machine translation, the video is treated as source language while the description to be generated is regarded as target language. In addition, the framework of “encoding-decoding” is employed to complete the conversion procedure from vision to language. The generated sentence with this solution is flexible in both sentence pattern and word picking, as well as in semantics since the pipeline maybe more close to expression habits of human. Moreover, the success of deep learning offers another opportunity of performance breakthrough for video description. Usually, the convolutional neural networks (CNN) are employed to extract feature of video for encoding, and then the visual feature is fed into recurrent neural networks (RNN) for decoding and generating candidate words one by one [5, 27, 28, 46]. Besides, more complicated hybrid models are further developed, combined with attention mechanism [6, 12, 16], visual attributes [8, 29, 52] and novel training strategies [9, 13] based on the pipeline, and the quality of generated sentence are also further improved.

However, different from image, the video contains not only static visual feature but the more important motion feature. Generally speaking, the special long-short term memory (LSTM) [19] network is frequently used to model the sequence of CNN feature for the motion feature. In the pipeline, the visual feature is generally facing forward propagation directly in LSTM networks, causing that the final representation is insufficient abstractness and poor representative ability, as well as the inadequate semantics in generated sentence in current models because a single LSTM unit possesses limited memory and the feature lacks enough non-linear transformation. Additionally, during decoding for candidate sentence, each word is predicted directly from LSTM at each time step, affecting the prediction accuracy of word in that the similar limited representative ability of language feature.

Facing the challenges, a double-channel language feature mining model is proposed for video description in this work (as shown in Fig. 1). For more abstract and discriminative representation of visual content and language at each time step, the output of sequential network is transformed again for motion feature extraction and candidate sentence generation. Concretely, the CNN feature of each frame is fed to the bottom LSTM firstly, and the output is sent to the added deep transformation layer (DTL) for visual representation enhancing, rather than directly to the top LSTM. For language modeling, in addition to the added DTL between the stacked two LSTMs, another DTL is appended on the top LSTM for language representation enhancing. Also, a DTL is embedded into the branch for extra language feature mining to further strengthen the power of language representation. Besides, the method of sequence cooperative decision [41, 42] is employed to fuse the shallow module (the top module in Fig. 1) and the deep module (the bottom module in Fig. 1)



**Fig. 1** The framework overview of the proposed double-channel language feature mining model (In visual encoding stage, there are  $t1$  time steps, which indicates that  $t1$  frames are fed to the network. During language decoding,  $t2 - t1 - 1$  ( $l$ ) time steps are contained. The  $LSTM_{lm}'$  is the language feature mining branch, and DTL is the deep transformation layer (a fully connected layer))

for boosting the accuracy of word prediction. In each module, the element-wise product strategy is employed for feature fusion. Finally, the outputs of the two modules are fused with weight average. In conclusion, the main contributions of this work includes the below aspects.

- A double-channel language feature mining model is developed in this work, where the usually employed two stacked LSTMs are disassembled by embedding an extra deep transformation layer (DTL) between them. The visual feature and language feature are transformed deeply with multiple DTLs after they are modeled by sequential network, enhancing the power of abstractness and representation.
- For comprehensive information, the scores from shallow module and relatively deep module are incorporated by sequence cooperative decision method, boosting the accuracy of word prediction. In each module, an early deep sequential fusion is employed with element-wise product operation, while the output probabilities from the two modules are fused with late deep sequential fusion with weight average.
- Experiments are implemented and ablation study is conducted on two public datasets including Youtube2Text and MSR-VTT2016, and competitive results compared to the state-of-the-art models are achieved, demonstrating the effectiveness of the proposed model. Additionally, the proposed method can be transferred to other frameworks easily, offering another solution of performance improvement for vision-language tasks.

The rest of this work is organized as follows. Section 2 introduces the related works about visual captioning. The motivation, proposed methods including visual and language deeply transforming and double-channel language mining are described in detail in Section 3. The

experimental results and discussion are presented in Section 4. Finally, Section 5 concludes the paper and prospects the future works.

## 2 Related works

Translating a video into natural language based on the visual content has been researched for decades. In early works, the researchers learn from the observation that there are many invariant sentence structures are frequently used in daily communication, and the techniques of computer vision such as object detection, pedestrian recognition and action classification are employed for the semantic ontologies and actions, following that the corresponding words are filled into the predesigned sentence templates [17, 23]. The framework is convenient and possesses high accuracy of predicted words. However, the generated sentence patterns are always inflexible and rigid, and the semantics and readability are greatly limited because of the fixed templates. In the other hand, another solution of semantic transferring is proposed for improving the flexibility of generated sentence. A query database including as many “video-description” pairs as possible is collected firstly, then the similar language components for the query video are retrieved with visual retrieval technique and the new description sentence is recomposed [51]. However, the method depends on the query database too much, and when there are no similar query pairs, great deviations between the generated sentence and the real video content may occur. Actually, the method based on semantic transfer is a coarse granularity of semantic entity division, which leads to the great limitation of the entities recombining, and then the quality of generated sentence will be affected.

Afterwards, the framework of “encoding-decoding”, which is for machine translation, is employed to generate sentence for video. The techniques in computer vision are usually used to extract visual feature for encoding, while the RNN networks is frequently employed to predict word one by one for decoding. This method divides semantic entities from smaller granularity (word-level), and the accuracy of predicted word, richness of semantics and flexibility of sentence pattern are all improved greatly. The deep learning offers another chance to boost the performance of model for video description since more abstract and powerful representation can be extracted. Nowadays, the framework of CNN+LSTM has been the most popular selection for the task [5, 27, 28, 46]. For modeling more accurate relation between visual region and language, the attention mechanism is introduced into the framework, where the visual information is selectively fed to the next time step according to the current prediction. This method reduces or even removes the unrelated or weakly related visual information that restrain the accuracy of word prediction at each time step, improving the quality of the generated sentence [6, 12, 16]. On the other hand, the visual attributes are also frequently collected from the references for different visual content that treated as specific visual objects or entities, and the attributes are detected in the video firstly, following the embedding of corresponding words in the sentence during decoding [29, 52]. Besides, the more effective memory unit such as Transformer is designed for parallel optimization and long-term dependency, offering another solution for image and video description [20, 44]

The methods mentioned above present effective attempts in the using of visual information. However, little attention has been paid to the processing and application of visual and language features. In this work, a double-channel language feature mining based video description model is developed. In detail, a deeper transformation layer (which is abbreviated as DTL) is implemented on sequential network to filter of the visual and language

features to enhance the abstract and representative ability, so as to improve the accuracy and semantics of generated sentence. Furthermore, the sequential fusion method [41, 42] is employed to further boost the accuracy of word prediction, in which the benchmark model without deeper transformation layer and the model with the layer are incorporated and each word is predicted according to outputs from the two models at each time step, enhancing the robustness of the whole generating framework.

### 3 Proposed DC-LFM model

#### 3.1 Motivation

Supposed that there are  $m$  frames in a video, and the frame sequence is denoted as  $\{f_1, f_2, \dots, f_m\}$ . And the feature sequence is  $\{v_1, v_2, \dots, v_m\}$  after that the frame sequence is transformed by a pre-trained CNN model. In current popular models such as LSTM-YT [47], S2VT [46] and deep fusion [48], the architecture with two stacked LSTMs is usually employed as the basis. In the pipeline, the CNN feature  $v_t$  of the frame  $f_t$  is firstly given to the bottom LSTM as  $x_t^1$  at the  $t$  time step in the S2VT model, and the output  $h_t^1$  of the hidden state of the bottom LSTM is then fed to the top LSTM as the  $x_t^2$ . When all the frames are exhausted, the word feature  $w_i$  of the  $i_{th}$  word in a sentence with length  $l$  and the hidden output from the bottom LSTM are both sent to the top LSTM at the  $m + i$  time step.

However, the output (hidden state) from the bottom LSTM layer is fed to the top LSTM layer directly at each time step, which indicates that the top LSTM layer at current time step is equivalent to state of the next time step of the bottom LSTM layer. As an example in Fig. 2, LSTM<sup>2</sup>\_1 in the top LSTM layer and LSTM<sup>1</sup>\_2 in the bottom LSTM share the same input which is from LSTM<sup>1</sup>\_1. It indicates that the top LSTM is just an extension of the time step and does not play a practical role in the model. For visual feature encoding, the method may limit the power of the final visual representation since there lacks enough linear and non-linear deep transformation. In the same way, during word prediction, the output of hidden state is employed to send to a prediction layer (a fully connected layer) directly, resulting in poor representation of language feature.

Facing the problem, a DTL layer is embedded between the two stacked LSTMs during visual feature encoding, where the output of hidden state from bottom LSTM is fed to the DTL directly, then the output of DTL is given to the top LSTM for further non-linear transformation. The module is marked as DT-MV for simplicity. Regarding to the stage of language modeling, the DTL in visual module is reserved since the two modules share the same architecture actually, and another DTL is appended on the top LSTM layer to enhance

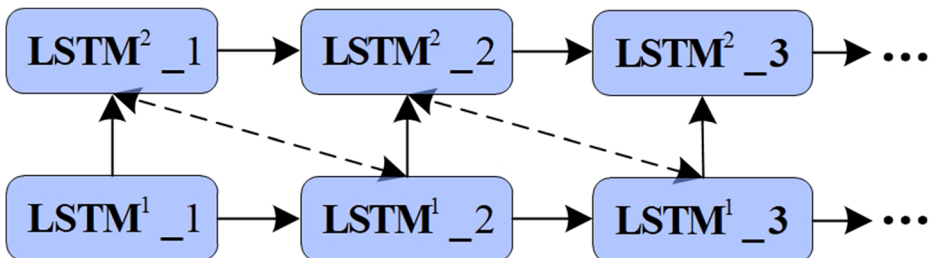


Fig. 2 The two stacked LSTMs architecture

the language representation (the module is named DT-ML for simplicity). Additionally, for mining richer language information, an extra LSTM is employed which just models the language. Same as the above mentioned language modeling, another DTL component is appended on the LSTM layer, and the output of the LSTM is fed to the DTL directly, then the transformed feature from the DTL is employed as the extra language representation, instead of the output from LSTM (the module is denoted by DT-LFM). The outputs of DT-ML and DT-LFM are fused with element-wise product operation. Then the fused feature is given to another DTL for further enhancing the abstractness of the final representation (where the fusion model is marked as DT-DFM). However, in consideration of that more abstractness may result in more details missing [40], the proposed DT-DFM and the model without any DTL (denoted as B-DFM) are further integrated. In detail, the output probabilities DT-DFM and B-DFM are fused by element-wise addition for word prediction at each time step.

### 3.2 Deeper transformation component of visual encoding

In most models such as deep fusion [48], the S2VT framework [46] is employed as the basis and another branch for language feature mining is combined to supplement more language information, improving the coherence and semantics of generated sentence. Additionally, the model lowers the parameter scale of the modules for visual feature dimensionality reduction and language feature embedding respectively, enhancing the efficiency of the model. In the pipeline, the CNN feature  $v_{t1}$  of the frame  $f_{t1}$  is firstly given to the bottom LSTM as  $x_{t1}^1$  at the  $t1$  time step in the S2VT model, and the output  $h_{t1}^1$  of the hidden state of the bottom LSTM is then fed to the top LSTM as the  $x_{t1}^2$ . When all the frames are exhausted, the word feature  $w_{t2}$  and the hidden output from the bottom LSTM are both sent to the top LSTM at the  $t2(t2 > t1)$  time step.

In the whole procedure, there are no other extra linear or non-linear transformation function between the two LSTM layers, and the temporal feature from the bottom LSTM is fed to the top LSTM directly. During extraction of visual motion feature, the output  $h_{t1}^1$  from the bottom LSTM is as  $x_{t1}^2$  and given to the top LSTM for initializing its memory cell at the first time step. While at the following time steps, the top LSTM continuously receives the output of the bottom LSTM. Generally, the output  $h_{t1}^2$  from the top LSTM at the  $t1$  time step is equivalent to the output  $h_{(t1+1)}^2$  of the bottom LSTM at the  $t1 + 1$  time step. It indicates that the top LSTM is just an extension of the time step and does not play a practical role in the model.

In order to enhance the representative ability of sequence feature and make the visual features for the following language model from the two LSTMs in different feature spaces, a deeper transformation component for visual encoding is proposed and the architecture is shown in Fig. 3. In detail, the output from the bottom LSTM is sent to a deeper transformation layer at each time step, and the sequence feature is then fed to the top LSTM after non-linear transformation. In the bottom LSTM, the visual feature at the  $t1$  time step is just as  $h_{t1}^1$  during forward propagation. And the visual motion feature  $v_B$  contains more motion details due to the fact that there is no extra linear and non-linear transformation layer, and thus the feature has the more power ability of fine-grained representation for the actions, scene transformations. But the generalization ability of the feature is not satisfactory. However, the feature space of the output  $x_{t1}^2$  from the DTL at the  $t1$  time step in the top LSTM has been transferred after non-linear transformation. Consequently, the final output  $v_T$  possesses stronger generalization ability, though part of details may be lost. When all frames are exhausted, the  $v_B$  and  $v_T$  will be fed into the language model as  $h_{(m+1)}^1$  and  $h_{(m+1)}^2$

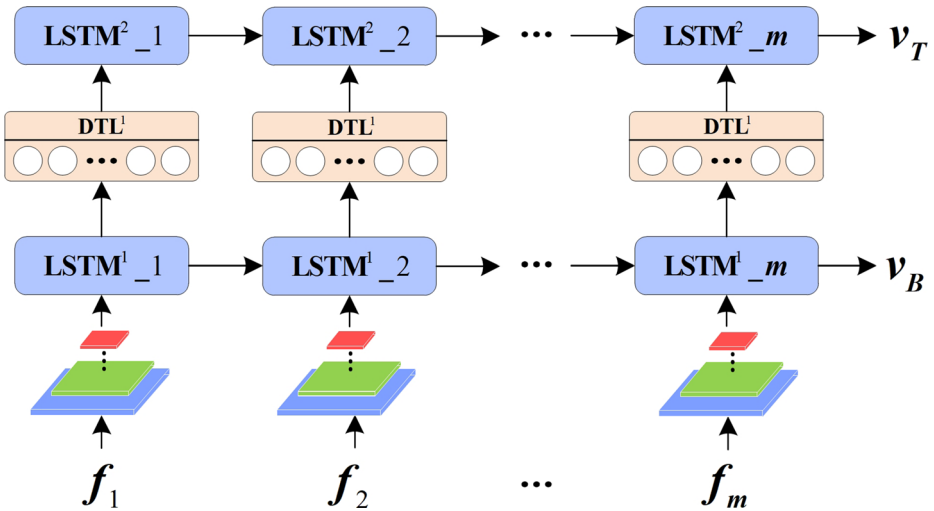


Fig. 3 Deeper transformation component for visual encoding (DT-MV)

respectively at the subsequent  $m + 1$  time step, making the model not only describe the video in detail but also enrich the semantics of description on the whole.

Let  $W_{dtl}^1$  and  $B_{dtl}^1$  denote the weight matrix and bias vector in  $DTL^1$  respectively, and the input  $x_{t1}^2$  for the top LSTM at the  $t1$  time step can be written as  $(x_{t1}^1)^1$ :

$$x_{t1}^2 = \begin{bmatrix} (x_{t1}^1)^1 \\ (x_{t1}^1)^2 \\ \vdots \\ (x_{t1}^1)^d \end{bmatrix} = DTL^1 \left( W_{dtl}^1 h_{t1}^1 + B_{dtl}^1 \right). \tag{1}$$

where the  $d$  is the number of hidden units in  $DTL^1$ , and  $DTL^1(\cdot)$  is the transformation function which includes the fully connected layer and non-linear activation layer. For simplification, the module is denoted as DT-MV (Deeper Transformation based Model for Visual encoding), which is for the motion feature encoding.

### 3.3 Deeper transformation component for language decoding

As presented in Fig. 4, the CNN feature sequence  $\{v_1, v_2, \dots, v_m\}$  of the video is transformed by DT-MV and the output is as the final visual representation and fed to the decoding stage at the  $m + 1$  time step. The “BoS” is the begin token of a sentence, and  $\{w_1, w_2, \dots, w_n\}$  is the embedding feature sequence of word in the sentence, where  $n$  is the number of words and  $w_i$  is usually obtained by “one-hot” method. During training, the number of the time steps in the whole network (including visual motion encoding and language decoding) is fixed (supposed that the number is  $ST$ ). The sum of the frames and words are calculated before encoding and decoding in consideration of that the lengths of different sentences may be different. If  $m + n + 1 \leq ST$  (where the  $m + n + 1$  time step is for the end token of sentence), the CNN feature of all frames and embedding feature of all words will take part in training; while if  $m + n + 1 > ST$ , the language feature should be in consideration firstly and all of them will be fed into the network, and the rest time steps



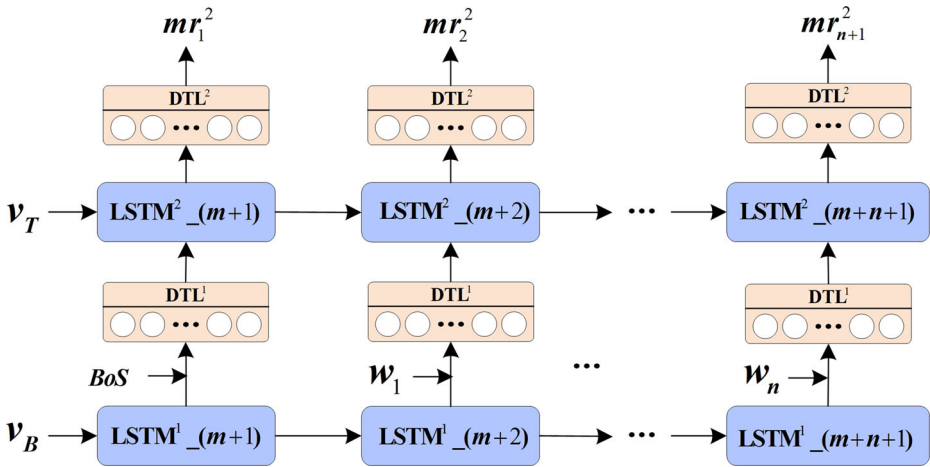


Fig. 4 Deeper transformation component for language decoding (DT-ML)

are left to the visual features due to their redundancy. At test time, the limitation for time step will be removed because the length of the generated sentence is unknown, and thus the feature of all frames will participate in the motion encoding in the network.

And the visual feature  $v_B$  continues to be forward propagation in time steps at the bottom LSTM layer. Meanwhile, the visual information is sent to the top LSTM according to the existing memory. On the other hand, the output of the bottom LSTM will be given to the top LSTM with the embedding feature for optimization of language model in traditional models. Though the language feature will be transformed again by the top LSTM, the final feature lacks powerful generalization ability as before caused by insufficient number of non-linear transformations. In addition, extra visual noises may be introduced into the model since there is visual information in the top LSTM, and then the accuracy of predicted words will be reduced, as well as semantics of the final generated sentence.

In this work, the DTL<sup>1</sup> layer is preserved during decoding stage. Meanwhile, a DTL<sup>2</sup> layer is appended on the top LSTM, where the visual representation is incorporated with the word embedding feature and fed to the DTL<sup>1</sup> as multi-modal feature at each time step. In this way, the multi-modal feature is more abstract in that it is transformed again. Particularly, the abstractness and representative ability of the language feature are both improved greatly after filtering with the DTL<sup>1</sup> layer. Besides, the visual noises are also filtered and suppressed since the output of the top LSTM is given to another DTL rather than the final classification layer for word prediction, improving the accuracy of word prediction and semantics of the whole sentence. Similarly, the DTL<sup>2</sup> is also employed to conduct re-transformation of the multi-modal features for the final word decision and further enhancing the generalization ability of the model.

For DTL<sup>1</sup>, the input is the multi-modal feature  $mr_{m+t2}^1$  can be obtained by the concatenation operation of the hidden output  $h_{m+t2}^1$  from the top LSTM and the word embedding feature  $w_{t2}$  at the  $m + t2 (1 \leq t2 \leq n + 1)$  time step. And the formula is:

$$mr_{m+t2}^1 = \text{Con}(h_{m+t2}^1, w_{t2}), \tag{2}$$



where the function  $\text{Con}(\cdot)$  is for feature concatenation. While for  $mr_{m+t2}^2$ , it can be calculated by

$$mr_{m+t2}^2 = \text{DTL}^2(W_{dtl}^2 h_{m+t2}^2 + B_{dtl}^2), \tag{3}$$

in which  $W_{dtl}^2$  and  $B_{dtl}^2$  stand for the weight matrix and bias vector respectively, and  $\text{DTL}^2(\cdot)$  is the deeper transformation function. For simplification, the module is denoted as DT-ML (Deeper Transformation based Model for Language decoding).

### 3.4 Language feature mining based on deeper transformation and deep fusion

In the module of DT-ML, the word embedding feature and visual representation are cooperated together for word prediction. The method bridges the visual content and language and makes the language module find the corresponding word in its memory according to visual information at each time step. However, extra visual noises may be introduced into the model, and the word sequence and original structure may be destroyed in certain since there is no independent sequential modeling of word feature. For reliving this limitation, another single LSTM layer is specially developed to model the language in deep fusion framework [48] and mine the latent language sequential feature. However, the word feature is just transformed with a single LSTM and has the similar trouble in DT-ML module, limiting the representative ability of feature. On the other hand, if multi-layer LSTM network is implemented, the parameters cannot be optimized sufficiently caused by gradient dispersion.

The deeper transformation method is also employed to meet the above challenge in this work. As shown in Fig. 5, the output at each time step is transformed and filtered again with  $\text{DTL}'_{lm}$  rather than word prediction directly after the modeling of word feature in a LSTM layer. At the  $m + t2$  time step, the output  $lr_{m+t2}$  of  $\text{DTL}'_{lm}$  can be calculated by

$$lr_{m+t2} = \text{DTL}'_{lm}(W_{dtl}^{lm} h_{m+t2}^{lm} + B_{dtl}^{lm}), \tag{4}$$

where  $W_{dtl}^{lm}$  and  $B_{dtl}^{lm}$  represent the weight matrix and bias vector in  $\text{DTL}'_{lm}$  layer respectively. For the convenience of presentation, the module is abbreviated as DT-LFM (Deeper Transformation based Language Feature Mining module).

The DT-LFM is just for modeling the language, which makes the model more sensitive to the internal structure and gets more accurate semantic information in sentences. But the module is not directly used for candidate sentence generation. On the other side, although

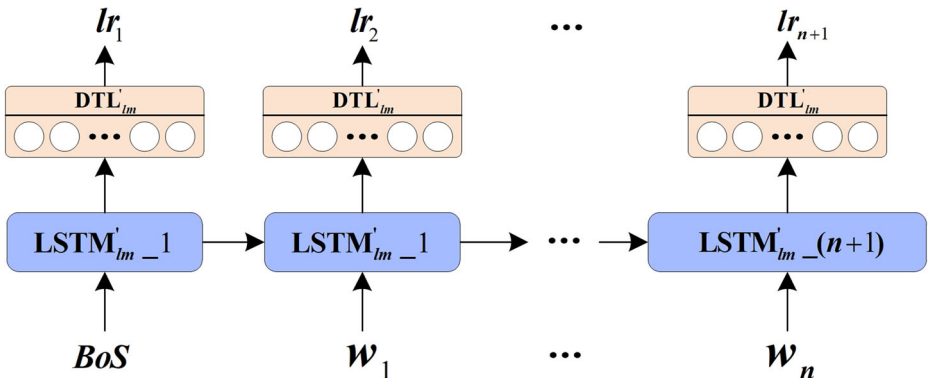


Fig. 5 Deeper transformation based language feature mining module (DT-LFM)

the DT-MV and DT-ML have the ability of sentence generation, the accuracy of word picking is unsatisfactory because of interference of the left visual information in DT-ML. Given all that, the three modules including DT-MV, DT-ML and DT-LFM are incorporated subtly to exploit their advantage and compensate each other. Also, another DTL is employed on the top of the model to calculate the probability score at each time step for word prediction.

As presented in Fig. 6, the multiplication operation is employed when the three components are combined, instead of the addition operation in traditional models like deep fusion [48]. At the  $m + t2$  time step, the outputs  $mr_{m+t2}^2$  and  $lr_{m+t2}$  from DT-ML and DT-LFM respectively are fused by element wise multiplication, and the output  $rf_{m+t2}$  can be written as

$$rf_{m+t2} = mr_{m+t2}^2 \otimes lr_{m+t2}. \tag{5}$$

Then the product is fed into  $DTL_p$  to conduct feature filtering and calculate the probability score for word decision. The module is marked as DT-DFM (Deeper Transformation based Deep Fusion Module) for simplification.

### 3.5 Double-channel fusion based on language feature mining

The DTL is employed in all the three components including DT-MV, DT-ML and DT-LFM, as well as the module DT-DFM, to enhance the abstractness of visual motion feature and language feature, and boost the semantics of candidate sentence generated by the final DT-DFM model. However, the more abstract of features actually indicates that more details maybe lost in the process of more linear and non-linear transformations. In contrast, the overall performance maybe limited in the model without multiple DTL (which is denoted as B-DFM), but more details will be remembered and may superior to the DT-DFM on the accuracy and appropriateness of word picking. Then the DT-DFM and B-DFM are incorporated to reconcile the contradiction. The sequence cooperative decision method [41, 42] is employed to predict the word at each time step, where the output probabilities of the two models are fused with weighted average.

As shown in Fig. 7, the output probability vector  $p_{t2}$  from  $DT-DFM_{t2}$  and the  $p'_{t2}$  from  $B-DFM_{t2}$  are fused by element wise weighted average at the  $m + t2$  time step, and the word corresponding to the position of the maximum fusion probability is the final prediction for the current time step.

where, for the fusion probability vector, the following formula is employed.

$$p_{t2}^f = \lambda_1 p_{t2} + \lambda_2 p'_{t2}, \tag{6}$$

in which the  $\lambda_1$  and  $\lambda_2$  are the harmonic factors, and they conform the constrain of  $\lambda_1 + \lambda_2 \triangleq 1$ . The fused model is denoted as DC-LFM (Double-Channel based Language Feature Mining model) for convenience.

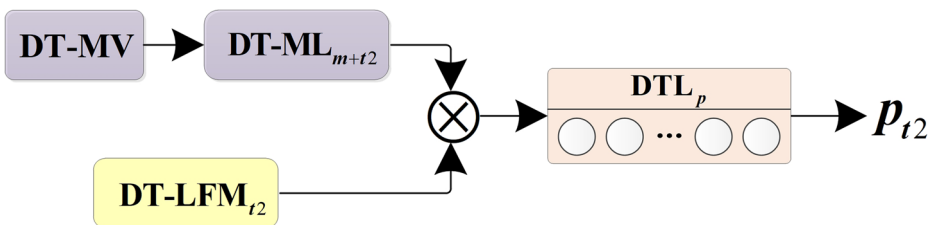


Fig. 6 Deeper transformation based deep fusion model

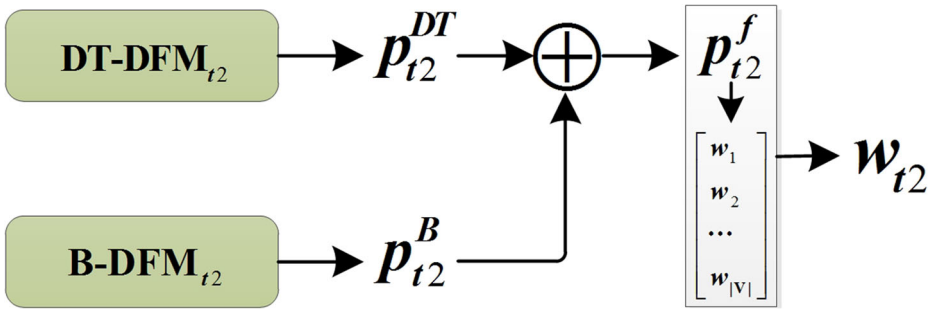


Fig. 7 Double-Channel based language feature mining model (DC-LFM)

Let  $O$  denote the objective function of the whole model, and it can be written as

$$O = \min_{\Theta} \mathcal{L}((V, S); \Theta), \tag{7}$$

where  $\mathcal{L}$  stands for the loss function, and  $\Theta = (\theta_{md}, \theta_{bd})$  is the parameter set, in which  $\theta_{md}$  and  $\theta_{bd}$  represent the parameter sub set of DT-DFM and B-DFM respectively. The  $V$  is the CNN feature set of videos, while the  $S$  indicates the reference sentence set. The objective of the whole model is to minimize the  $\mathcal{L}$  according to the algorithm of back propagation through time (BPTT). At the  $k$ th iteration, the  $\mathcal{L}$  can be calculated with

$$\mathcal{L}_k = \frac{1}{N} \sum_{i=1}^N \sum_{j=0}^{|S_i|} \log p(w_{j+1}|(E(V_i), w_{1:j}); \Theta), \tag{8}$$

in which the  $N$  is the number of samples in one iteration (batch size), and  $S_i$  is the length (number of words) of the reference sentence corresponding to the  $i$ th video. When  $j = 0$ , the  $w_0$  denotes the token of beginning of sentence (“BoS”). The  $V_i$  stands for the CNN feature sequence of the  $i$ th video, and the function  $E(\cdot)$  represents the module of DT-MV which is for visual motion encoding.

During training, the error signal from loss function  $\mathcal{L}$  will be back propagated to DT-DFM and B-DFM simultaneously for updating the parameter sets of  $m_d$  and  $b_d$ . In this way, the two modules will complement each other. The B-DFM can be optimized sufficiently by more effective error signals though there are no extra DTLs since the gradients are computed by both of B-DFM and DT-DFM. On the other side, the DT-DFM refers the visual and language details from B-DFM. So the generated sentence not only concerns the details but also semantics in the whole.

At test stage, the function of Softmax is employed to calculate the output probability vectors of DT-DFM and B-DFM, then the fusion score vector is computed by (6) and the corresponding word is picked for prediction. At the  $t_2$  time step, the formula for the fusion probability  $(p_{t_2}^r)^{(DT/B)}$  that the  $r$ th word in the vocabulary belongs to the current state is

$$(p_{t_2}^r)^{(DT/B)} = \frac{\exp((\theta_r)^T C_r^{DT/B})}{\sum_{s=1}^{|\mathbb{V}|} \exp((\theta_s)^T C_s^{DT/B})}, \tag{9}$$

where the  $(\theta_r)^T$  is the weight vector belonging to the  $r$ th hidden unit in classification layer, and  $C_r^{(DT/B)}$  is the output of the unit in DT-DFM or B-DFM module, while the  $|\mathbb{V}|$  is the size (number of words) of vocabulary  $\mathbb{V}$ . When the calculation of fusion probability vector is completed, a dummy label corresponding the maximum score is mapped to the appropriate

word in vocabulary and as the final predicted word at the current time step. Given that  $G(\cdot)$  is the mapping function, the formula for word selecting can be written as

$$w_{t2} = G(\max(p_{t2}^f) \mapsto \mathbb{V}). \quad (10)$$

Besides, the beam search algorithm which belongs to a heuristic search method is employed to further improve the accuracy of word prediction and semantics of generated sentence in this work.

## 4 Experiment

Multi group experiments are conducted to evaluate the effectiveness of the proposed model. The experimental environment and configuration are introduced firstly in this section, as well as the used datasets and evaluation metrics. Then the experimental results are presented and the generated sentences are analyzed subjectively. Finally, the statistical results on a few popular metrics are provided, indicating the effectiveness and superiority of the proposed model.

### 4.1 Experimental setting

The popular deep learning framework Caffe [21] is employed to implement the benchmark and our proposed model. To speed the convergence of the model, 2 NVIDIA TITAN X GPUs are used for training and test. The CNN model is pre-trained firstly on ImageNet [35] to prevent the model to stick to over fitting, and then the model is continue fine-tuned on MSCOCO2014 [26] with LRCN framework [14] to make the model sensitive to the sentence patterns and frequently used words. The jointly modeling strategy is employed to avoid local optimum during fine-tuning of the model, where the visual deep model and language model are optimized together. Afterwards, the visual feature model is taken out separately to extract the CNN feature of video frame. In this work, GoogLeNet [39] and ResNet152 [18] are used as the visual models, and the CNN feature from the last pooling layers with the dimension of 1024 and 2048 respectively in the two models are as the visual representation of the video frame. In order to achieve a good trade-off between redundancy of video frames and sufficiency of visual content for video pre-processing, we follow the practice in [46, 47]. Concretely, one frame is sampled every 10 frames (1 frame in every 10 frames) in each video. Then the sampled frames are fed to GoogLeNet or ResNet152 for visual features and as the input for LSTM.

Three models are developed based on the deep fusion framework [48] in this work. In the first place, the benchmark model (B-DFM) is implemented for comparison, and the settings are similar to deep fusion, where the dimension of visual feature is reduced to 512 before it is fed to LSTM network and the dimension of embedding feature of each word is set to 300. Secondly, the evaluation of proposed DT-DFM is conducted. The number of output in every used DTL is set to the same as the output dimension of LSTM connected to it for dimension consistency, which indicates that the number of output in each used DTL is 1024. Finally, the DC-LFM which is consists of B-DFM and DT-DFM is implemented, where the harmonic factor for fusion is empirically set to 0.5.

At the training stage, the number of time steps including visual motion feature encoding and language decoding is fixed to 80 ( $ST = 80$ ) since the number allows the model to fit multiple videos in a single mini-batch and is helpful to speed up the whole training process and convergence, which also follows the practice in [46, 47] actually. In the view of visual

data redundancy, all the words in reference should be sent to LSTM, and the rest time steps are for visual motion feature encoding. Suppose that the length of reference is  $s_k$ , and the number of the rest time steps is  $80 - s_k$ . Let  $l_k$  be the number of video frames after sampling, and when  $l_k > 80 - s_k$ , the frames exceeding the limit will be truncated. While if  $l_k < 80 - s_k$ , the extra time steps will be fed with  $\langle pad \rangle$ . At test time, the limitation on time steps of LSTM is removed, which indicates that all frames will take part in visual motion feature encoding and the following word prediction.

In the whole pipeline of model optimization, the frequently used stochastic gradient descent method is employed to update parameters. The learning rate is initially set to  $2 \times 10^{-3}$ , which will be reduced to 0.5 times every 40 K iterations in the following training. For Youtube2Text dataset [7, 17], the maximum of iteration is 200K, while the value is set as 300K on MSR-VTT2016 dataset [55] due to its larger scale samples. The Dropout is employed to prevent the model to fall into over fitting on the both datasets, and the drop ratios are assigned to 0.5 and 0.3 according to our empirical observation. Besides, the beam search algorithm is used to further improve performance, and the size of searching pool is set to 5, following the practice in other popular works.

## 4.2 Evaluation dataset and metric

Two popular public datasets including Youtube2Text and MSR-VTT2016 are employed to evaluate the proposed model. As for Youtube2Text, there are 1,970 video clips and the corresponding reference sentences annotated by human. We follow the practice in S2VT [46], where 1200 clips and references are as training set, and 670 video clip-reference pairs are for testing. The rest pairs are for validation. Regarding to MSR-VTT2016 dataset, 10,010 video clips are contained, and each clip has 20 references. According to using protocol, 6,513 clip-reference pairs are for model optimizing, while 497 pairs are used to find hyper parameters, and the rest pairs are used as test set. During training, the pairs in validation set will be put back to the training set when the model converges, then the model is fine-tuned on the new training set until iteration reaches to the convergence point. The vocabulary used in deep fusion model [48] is employed in this work, and the size is 72,700.

Four metrics including BLEU [30], METEOR [4], ROUGE-L [25] and CIDEr [45] are employed to conduct objective evaluation on the proposed model. For BLEU, the number of  $n$ -Gram ( $n \in (1, 2, 3, 4)$ ) matches between references and candidate sentences is counted, then the ratio of the number and the counts of  $n$ -Gram in candidate sentences is calculated. Generally speaking, the higher the ratio, the higher quality of generated sentences. As for METEOR metric, the maximum of matches is computed firstly according to exactly match, synonymous match and word-stem match of the words in references and candidate sentences. Afterwards, the match with the least number of intersections in the two matches in order is selected to generate match set. Then, the ratio between size of the set and number of words in references is computed as recall rate, meanwhile the ratio between size of the set and number of words in candidate sentences is as precision rate. Finally, the harmonic mean value is calculated as the score. Different from METEOR, the longest common subsequence is employed to calculate the recall and precision in ROUGE-L. In contrast, inspired by “consensus” concept, different  $n$ -Gram tuples are assigned different weights to mark their significance in CIDEr metric. And the matching degree between references and candidate sentences is measured according to cosine distance. In addition, a few examples generated by our proposed model are presented in this work for subjective evaluation. The advantage and limitation of our model are discussed and analyzed in the way of providing comparison of references and generated sentences.

### 4.3 Experimental result and discussion

A number of generated sentences are in exhibition firstly in this section to demonstrate the superiority of the proposed model according to comparison with the real visual content, reference and generated sentences by the benchmark model. Then the performance of the proposed model on different metrics with different CNN feature is presented. Finally, the comparison with other popular models is also shown and analyzed.

#### 4.3.1 Example and discussion

As shown in Fig. 8, a few examples from Youtube2Text dataset including the video clip-reference pairs, generated sentences with benchmark model (which is denoted as “B”) and proposed DC-LFM model (which is denoted as “P”) are in exhibition. From the intuitive



Ref: A basketball player basketting the ball.  
 A man is playing basketball.  
 A man shoots a basket.  
 The man is playing basketball.  
 The man runs and puts the ball into the basket.

**B: A man is kicking a basketball.**

**P: A man is playing basketball.**

(a)



Ref: A dog is swimming in a pool.  
 A dog is swimming in the swimming pool.  
 A dog jumps into a pool and swims.  
 A dog is diving in the pool.  
 A funny dog swimming in the pool.

**B: Two people are swimming in a pool.**

**P: A dog is swimming in a pool.**

(b)



Ref: A man is cycling in the wood.  
 A man is riding a bike over a ramp in the woods.  
 The man rode his bike into the woods.  
 A motorcyclist is doing jumps while riding through the woods.  
 A boy is riding bicycle in a forest.

**B: A man is running**

**P: A man is running through the woods.**

(c)



Ref: Several teams are playing soccer.  
 Men are playing soccer.  
 Players are playing a football.  
 People are playing soccer.  
 Some men are playing soccer.

**B: A soccer player is playing the ball.**

**P: Men are playing soccer.**

(d)



Ref: A woman is frying something.  
 A person is dropping battered food into a pot.  
 A cook puts some vegetables into a pot.  
 A lady is preparing food.  
 The lady added the batter dipped shrimp to the oil.

**B: A man cooking his unkn.**

**P: A person is adding ingredients into a bowl.**

(e)



Ref: A cat is jumping into a box.  
 A cat jumps into a box.  
 A cat is playing.  
 A cat jumps into a tall box.  
 A cat walking in the house.

**B: A cat is watching a box.**

**P: A cat is playing with a box.**

(f)

**Fig. 8** Examples of the reference (from Youtube2Text dataset), generated sentence with benchmark model and the proposed DC-LFM model (with ResNet152 feature)

**Table 1** Performance of the proposed model on Youtube2Text dataset (with GoogLeNet feature)

Model	B1	B2	B3	B4	METEOR	ROUGE-L	CIDEr
B-DFM	77.4	64.4	53.4	42.5	31.2	67.5	62.1
DT-DFM	75.8	63.4	53.2	42.8	31.6	66.8	66.7
DC-LFM	77.9	65.6	55.2	44.7	31.6	68.4	66.1

observation, the generated sentences with DC-LFM are more accurate and decent than that with the benchmark model in that there are more deviations in the sentences compared to the visual content. For instance, the “kicking” is used as predicate in Fig. 8a, but it is obviously inappropriate for the object (“basketball”). In contrast, the verb of “playing” is picked with DC-LFM. And in Fig. 8b, the predicate (“swimming”) and object (“pool”) are predicted accurately with the benchmark model. However, the model provides false subject. Comparatively, the DC-LFM model describes the video content exactly and completely. In addition, the generated sentences from DC-LFM are more comprehensive and richer semantics. As an example in Fig. 8e, the DC-LFM well describes not only the action (“adding”) of the subject, but also the object (“ingredients”) and whereabouts (“bowl”).

In comparison to reference sentences (which are denoted as “Ref”), it can be observed that several generated sentences with DC-LFM are more detailed description and accurate. However, partial visual objects cannot be predicted correctly with the proposed model. For example, the action (“riding”) of subject and object (“bicycle”) are not detected in Fig. 8c. Besides, compared with references, the generated sentences are not flexible enough in sentence pattern and word using. The possible reason is that the insufficient training samples lead to the model sticking to over fitting. On the other side, lots of visual details maybe lost due to multiple linear and non-linear transformations in CNN and LSTM models, resulting in the loss of certain flexibility in the sentences though generalization ability is improved of the model.

#### 4.3.2 Statistical result and analysis

The performances of B-DFM, DT-DFM and DC-LFM on Youtube2Text and MSR-VTT2016 datasets with GoogLeNet and ResNet152 features are presented in this section. The performance on each metric is shown in Table 1 on Youtube2Text with GoogLeNet feature. It is obvious that the performances on B4, METEOR and particularly CIDEr are all improved when DTL is appended on the benchmark model. However, the B1, B2 and B3 are decreased a little. It indicates that the added DTL makes the feature more abstract and generated sentences possess richer semantics, but yields to the accuracy of word prediction since the added non-linear transformation may lead to the loss of a few details. And when the DC-LFM is employed, the performances on all metrics but CIDEr are improved greatly. It shows that the B-DFM and DT-DFM can complement each other and better the semantics and coherence of generated sentences.

**Table 2** Performance of the proposed model on Youtube2Text dataset (with ResNet152 feature)

Model	B1	B2	B3	B4	METEOR	ROUGE-L	CIDEr
B-DFM	80.6	68.9	59.0	48.5	33.7	70.7	77.3
DT-DFM	78.9	67.1	56.9	46.4	34.0	70.0	79.8
DC-LFM	80.9	69.7	60.0	49.8	34.8	71.3	82.5



**Table 3** Performance of the proposed model on MSR-VTT2016 dataset (with GoogLeNet feature)

Model	B1	B2	B3	B4	METEOR	ROUGE-L	CIDEr
B-DFM	76.5	61.6	48.2	36.8	26.3	58.2	39.6
DT-DFM	76.1	61.5	48.2	37.0	26.7	57.9	40.1
DC-LFM	76.4	61.9	48.6	37.3	26.7	58.5	40.7

Similarly, when ResNet152 feature is used, the performances on METEOR and CIDEr are both improved with DT-DFM (as shown in Table 2), but the BLEU and ROUGE-L are decreased. However, the performances on all metrics are improved when B-DFM and DT-DFM are fused, particularly, the CIDEr reaches to 82.5.

On MSR-VTT2016 dataset, the performance of each model on each metric is similar to that on Youtube2Text (Tables 3 and 4). DT-DFM possesses better performance on B4, METEOR and CIDEr than B-DFM, but worse on B1, B2 and ROUGE-L. While the DC-LFM overcomes the limitation and takes advantage of the merits of B-DFM and DT-DFM, boosting performance on all metrics. When performance comparison is conducted with different CNN features, DT-DFM model with ResNet152 feature achieves better performance on BLEU and ROUGE-L metrics than B-DFM but yields to that on CIDEr which usually reflects semantics. The trend is different to that the performance on Youtube2Text dataset with ResNet152 and GoogLeNet features. It probably because the MSR-VTT2016 is more clean and has less visual and language noises. The more abstract CNN feature from ResNet152 may lead to certain over fitting when multiple DTL is added on B-DFM. And thus generalization ability of the model is reduced, affecting the semantics of generated sentences. However, the performance of DC-LFM is not limited since the two models of B-DFM and DT-DFM form information complementary, and both of coherence and semantics are effectively enhanced.

In Tables 5 and 6, the performance comparison to the other state-of-the art models is provided. It can be seen that the performance of DC-LFM (with ResNet152) outperforms most of current methods on various metrics on Youtube2Text dataset, particularly it exceeds the most popular C3D fc7+pool4 model [31] by 11.5 on CIDEr. Additionally, the proposed model reaches to the almost the similar performance compared to the h-RNN model [58] on BLEU, and outperforms the C3D fc7+pool4 by 1.6. It proves that the generated sentences with DC-LFM possess richer semantics and more accuracy, as well as the narrow semantic gap compared to references. Besides, competitive results are also obtained with DC-LFM compared to other popular models on MSR-VTT2016. On the whole, the generated sentences with DC-LFM give consideration to both coherence and semantics. However, the advantage is not obvious compared with other models on MSR-VTT2016. One of the most reason is that multi-modal information including visual content, voice is integrated employed to boost model performance. As an example, the information scores of vision,

**Table 4** Performance of the proposed model on MSR-VTT2016 dataset (with ResNet152 feature)

Model	B1	B2	B3	B4	METEOR	ROUGE-L	CIDEr
B-DFM	78.2	63.8	50.2	38.5	27.4	59.5	44.6
DT-DFM	78.5	64.8	51.4	39.4	27.4	59.6	43.7
DC-LFM	79.4	65.8	52.5	40.4	27.8	60.5	45.9

**Table 5** Performance comparison with the state-of-the-art methods on Youtube2Text dataset

Method	B1	B2	B3	B4	METEOR	CIDEr
FGM [43]	–	–	–	13.7	23.9	–
LSTM-YT [47]	–	–	–	33.3	29.1	–
S2VT [46]	–	–	–	–	29.8	–
MM-VDN [54]	–	–	–	37.6	29.0	–
HRNE [27]	79.2	66.3	55.1	43.8	33.1	–
SA [57]	80.0	64.7	52.6	41.9	29.6	51.7
LSTM-E [28]	78.8	66.0	55.4	45.3	31.0	–
Boundary-aware encoder [5]	–	–	–	42.5	32.4	63.5
GRU-RCN [3]	–	–	–	43.3	31.6	67.8
h-RNN [58]	81.5	70.4	60.4	49.9	32.6	65.8
C3D fc7+pool4 [31]	–	–	–	48.2	34.5	71.0
LSTM-GAN [56]	–	–	–	42.9	30.4	–
Li et al. [24]	–	–	–	48.0	31.6	68.8
RecNet <sub>local</sub> (SA-LSTM) [49]	–	–	–	52.3	34.1	80.3
MS-RNN(R) [38]	82.9	72.6	63.5	53.3	33.8	74.8
TDCovED(R) [11]	–	–	–	53.3	33.8	76.4
GRU-EVE <sub>hft+sem</sub> (CI) [1]	–	–	–	47.9	35.0	78.1
DC-LFM(GoogLeNet)	77.9	65.6	55.2	44.7	31.6	66.1
DC-LFM(ResNet152)	80.9	69.7	60.0	49.8	34.8	82.5

voice, auditory and tuple are fused together to capture richer video representation in v2t navigator model [22]. While just the visual content is used in our work in consideration of pair comparison.

**Table 6** Performance comparison with the state-of-the-art methods on MSR-VTT2016 dataset

Method	B4	METEOR	ROUGE-L	CIDEr
ReBiLSTM [5]	33.9	26.2	–	–
LSTM-GAN (attention) [56]	36.0	26.1	–	–
Li et al. [24]	37.5	26.4	–	–
aLSTMs [16]	38.0	26.1	–	–
M3-IC [50]	38.1	26.6	–	–
MS-RNN(R) [38]	39.8	26.1	59.3	40.9
RecNet <sub>local</sub> (SA-LSTM) [49]	39.1	26.6	59.3	42.7
ruc-uva [15]	38.7	26.9	58.7	45.9
VideoLAB [33]	39.1	27.7	60.6	44.1
Aalto [37]	39.8	26.9	59.8	45.7
v2t navigator [22]	40.8	28.2	60.9	44.8
PickNet(V+L+C) [10]	41.3	27.7	–	44.1
TDCovED(R) [11]	39.5	27.5	–	42.8
GRU-EVE <sub>hft+sem</sub> (CI) [1]	38.3	28.4	–	48.1
DC-LFM (GoogLeNet)	37.3	26.7	58.5	40.7
DC-LFM (ResNet152)	40.4	27.8	60.5	45.9

**Table 7** The average time cost (ms) for each video captioning of the baseline model (B-DFM) and the proposed models (DT-DFM and DC-LFM) with ResNet152 feature

Model	Time consuming (ms)
B-DFM	57.81
DT-DFM	53.23
DC-LFM	101.78

### 4.3.3 Time cost of different models and analysis

Ablation study about the time cost of different models is conducted. For practice details, the models including B-DFM, DT-DFM and DC-LFM are tested on Youtube2Text dataset, and the running time of all sentences generated (670) are recorded. Then the average time consuming is as the cost for each video. The results with ResNet152 feature and GoogLeNet feature are as shown in Tables 7 and 8 respectively.

From the comparison, it can be observed that the DT-DFM consumes the least time no matter with ResNet152 feature or GoogLeNet feature. The reason is that the element-wise product operation saves more time compared to the concatenation operation in B-DFM though more layers (DTL) are appended on DT-DFM model. For DC-LFM model, the time cost is relative expensive, with exceeding both the other two models. However, the DC-LFM consists of B-DFM and DT-DFM, and the time consuming is still lower than the sum of the other two models regardless ResNet152 feature or GoogLeNet feature is employed.

## 5 Conclusion

Describing a video with natural language is interesting but challenging since the task involves not only computer vision but also natural language processing. The breakthrough of deep learning offers an opportunity to boost generated sentence quality, in particularly the framework of CNN+LSTM is the most popular solution for video description. However, the current works usually focus on refinement of visual content by constructing more reasonable mapping relationship between visual information and language to further improve performance, and the abstraction and representative ability of motion and language features are always ignored, leading to that the potentiality is not fully explored. In this work, deeper transformation layers are appended on both visual motion encoding stage and language modeling stage to deepen the model and enhance representative ability of visual and language features. Besides, the sequential cooperative decision method is applied on our proposed model to improve robustness. The relatively shallow B-LFM and the proposed deeper DT-LFM are incorporated into DC-LFM to predict the word collaboratively at each time step. Experimental results on Youtube2Text and MSR-VTT2016 demonstrate that the proposed model is more effective compared to not only the benchmark model but also the state-of-the-art methods. In the future work, the proposed method will be further improved

**Table 8** The average time cost (ms) for each video captioning of the baseline model (B-DFM) and the proposed models (DT-DFM and DC-LFM) with GoogLeNet feature

Model	Time consuming (ms)
B-DFM	57.81
DT-DFM	53.23
DC-LFM	101.78

and implemented on other advanced frameworks such as DenseVidCap [36]. Also, the proposed idea will be further extended and applied on a few other interesting multimedia tasks including data exchange [53], image segmentation [2, 59], object detection [32, 34] and image dehaze [60].

## References

1. Aafaq N, Akhtar N, Liu W, Gilani SZ, Mian A (2019) Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In: IEEE conference on computer vision and pattern recognition, pp 12487–12496
2. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(12):2481–2495
3. Ballas N, Yao L, Pal C, Courville A (2015) Delving deeper into convolutional networks for learning video representations. In: International conference on learning representations
4. Banerjee S, Lavie A (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Annual meeting of the association for computational linguistics workshop, pp 65–72
5. Baraldi L, Costantino G, Rita C (2017) Hierarchical boundary-aware neural encoder for video captioning. In: IEEE conference on computer vision and pattern recognition, pp 3185–3194
6. Bin Y, Yang Y, Shen F, Xie N, Shen H, Li X (2019) Describing video with attention based bidirectional lstm. *IEEE Trans Cybern* 49(7):2631–2641
7. Chen DL, Dolan WB (2011) Collecting highly parallel data for paraphrase evaluation. In: The 49th annual meeting of the association for computational linguistics, pp 190–200
8. Chen H, Ding G, Lin Z, Zhao S, Han J (2018) Show, observe and tell: attribute-driven attention model for image captioning. In: International joint conference on artificial intelligence, pp 606–612
9. Chen H, Ding G, Zhao S, Han J (2018) Temporal-difference learning with sampling baseline for image captioning. In: The AAAI conference on artificial intelligence, pp 6706–6713
10. Chen Y, Wang S, Zhang W, Huang Q (2018) Less is more: picking informative frames for video captioning. In: The European conference on computer vision, pp 367–384
11. Chen J, Pan Y, Li Y, Yao T, Chao H, Mei T (2019) Temporal deformable convolutional encoder-decoder networks for video captioning. In: Proceedings of the association for the advance of artificial intelligence, pp 8167–8174
12. Cho K, Courville A, Bengio Y (2015) Describing multimedia content using attention-based encoder-decoder networks. *IEEE Trans Multimed* 17(11):1875–1886
13. Ding G, Chen M, Zhao S, Chen H, Han J, Liu Q (2019) Neural image caption generation with weighted training and reference. *Cogn Comput* 11(6):763–777
14. Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition, pp 2625–2634
15. Dong J, Li X, Lan W, Huo Y, Snoek CG (2016) Early embedding and late reranking for video captioning. In: ACM conference on multimedia conference, pp 1082–1086
16. Gao L, Guo Z, Zhang H, Xu X, Shen H (2017) Video captioning with attention-based lstm and semantic consistency. *IEEE Trans Multimed* 19(9):2045–2055
17. Guadarrama S, Krishnamoorthy N, Malkarnenkar G, Venugopalan S, Mooney R, Darrell T, Kate S (2013) Youtube2text: recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: IEEE international conference on computer vision, pp 2712–2719
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
19. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 09(08):1735–1780
20. Iashin V, Rahtu E (2020) A better use of audio-visual cues: Dense video captioning with bi-model transformer. [arXiv:2005.08271v1](https://arxiv.org/abs/2005.08271v1)
21. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, et al. (2014) Caffe: convolutional architecture for fast feature embedding. In: ACM conference on multimedia, pp 675–678
22. Jin Q, Chen J, Chen S, Xiong Y, Hauptmann A (2016) Describing videos using multimodal fusion. In: ACM conference on multimedia conference, pp 1087–1091
23. Krishnamoorthy N, Malkarnenkar G, Mooney RJ, Saenko K, Sergio G (2013) Generating natural-language video descriptions using text-mined knowledge. The AAAI conference on artificial intelligence, pp 541–547

24. Li W, Guo D, Fang X (2018) Multimodal architecture for video captioning with memory networks and an attention mechanism. *Pattern Recognit Lett* 105:23–29
25. Lin CY, Och FJ (2004) Automatic evaluation of machine translation quality using longest common sub-sequence and skip-bigram statistics. In: Annual meeting of the association for computational linguistics, pp 21–26
26. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollr P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision, pp 740–755
27. Pan P, Xu Z, Yang Y, Wu F, Zhuang Y (2016) Hierarchical recurrent neural encoder for video representation with application to captioning. In: IEEE conference on computer vision and pattern recognition, pp 1029–1038
28. Pan Y, Mei T, Yao T, Li H, Rui Y (2016) Jointly modeling embedding and translation to bridge video and language. In: IEEE conference on computer vision and pattern recognition, pp 4594–4602
29. Pan Y, Yao T, Li H, Mei T (2017) Video captioning with transferred semantic attributes. In: IEEE international conference on computer vision, pp 984–992
30. Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Annual meeting of the association for computational linguistics, pp 311–318
31. Pu Y, Min MR, Gan Z, Carin L (2016) Adaptive feature abstraction for translating video to language. arXiv:1611.07837
32. Quan Q, He F, Li H (2020) A multi-phase blending method with incremental intensity for training detection networks. *Multimed Tools Appl*, in press. <https://doi.org/10.1007/s00371-020-01796-7>
33. Ramanishka V, Abir D, Huk PD, Subhashini V, Anne HL, Marcus R, Kate S (2016) Multimodal video description. In: ACM conference on multimedia conference, pp 1092–1096
34. Ren S, He K, Girshick R, Sun J (2016) Faster R-cnn: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
35. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, et al. (2015) Imagenet large scale visual recognition challenge. *Int J Comput Vis* 115(3):211–252
36. Shen Z, Li J, Su Z, Li M, Chen Y, Jiang Y-G, Xue X (2017) Weakly supervised dense video captioning. In: IEEE conference on computer vision and pattern recognition, pp 5159–5167
37. Shetty R, Laaksonen J (2016) Frame- and segment-level features and candidate pool evaluation for video caption generation. In: ACM conference on multimedia conference, pp 1073–1076
38. Song J, Guo Y, Gao L, Li X, Alan H, Shen H (2019) From deterministic to generative: multimodal stochastic rnns for video captioning. *IEEE Trans Neural Netw Learn Syst* 30(10):3047–3058
39. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, pp 1–9
40. Tang P, Wang H, Kwong S (2017) G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing* 225:188–197
41. Tang P, Wang H, Kwong S (2018) Deep sequential fusion lstm network for image description. *Neurocomputing* 312:154–164
42. Tang P, Wang H, Li Q (2019) Rich visual and language representation with complementary semantics for video captioning. *ACM Trans Multimed Comput Commun Appl* 15(2:31):1–23
43. Thomason J, Venugopalan S, Guadarrama S, Saenko K, Mooney R (2014) Integrating language and vision to generate natural language descriptions of videos in the wild. In: Proceedings of international conference on computational linguistics, pp 1218–1227
44. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: The conference on neural information processing systems, pp 5998–6008
45. Vedantam R, Zitnick CL, Parikh D (2015) CIDEr: consensus-based image description evaluation. In: IEEE conference on computer vision and pattern recognition, pp 4566–4575
46. Venugopalan S, Rohrbach M, Donahue J, Mooney R, Darrell T, Saenko K (2015) Sequence to sequence–video to text. In: IEEE international conference on computer vision, pp 4534–4542
47. Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney R, Saenko K (2015) Translating videos to natural language using deep recurrent neural networks. In: The 2015 annual conference of the North American chapter of the ACL, pp 1494–1504
48. Venugopalan S, Hendricks LA, Mooney R, Saenko K (2016) Improving lstm-based video description with linguistic knowledge mined from text. In: Conference on empirical methods in natural language processing, pp 1961–1966
49. Wang B, Ma L, Zhang W, Liu W (2018) Reconstruction network for video captioning. In: IEEE conference on computer vision and pattern recognition, pp 7622–7631
50. Wang J, Wang W, Huang Y, Wang L, Tan T (2018) M3: multimodal memory modelling for video captioning. In: IEEE conference on computer vision and pattern recognition, pp 7512–7520

51. Wei S, Zhao Y, Zhu Z, Nan L (2010) Multimodal fusion for video search reranking. *IEEE Trans Knowl Data Eng* 22(8):1191–1199
52. Wu Q, Shen C, Liu L, Dick A, Hengel A (2016) What value do explicit high level concepts have in vision to language problems? In: *IEEE international conference on computer vision*, pp 203–212
53. Wu Y, He F, Zhang D, Li X (2018) Service-oriented feature-based data exchange for cloud-based design and manufacturing. *IEEE Trans Serv Comput* 11(2):341–353
54. Xu H, Venugopalan S, Ramanishka V, Rohrbach M, Saenko K (2015) A multi-scale multiple instance video description network. [arXiv:1505.05914](https://arxiv.org/abs/1505.05914)
55. Xu J, Mei T, Yao T, Rui Y (2016) MSR-VTT: a large video description dataset for bridging video and language. In: *IEEE conference on computer vision and pattern recognition*, pp 5288–5296
56. Yang Y, Zhou J, Ai J, Bin Y, Hanjalic A, Shen H (2018) Video captioning by adversarial lstm. *IEEE Trans Image Process* 27(11):5600–5611
57. Yao L, Torabi A, Cho K, Ballas N, Pal C, Larochelle H, Courville A (2015) Describing videos by exploiting temporal structure. In: *IEEE international conference on computer vision*, pp 4507–4515
58. Yu H, Wang J, Huang Z, Yang Y, Xu W (2016) Video paragraph captioning using hierarchical recurrent neural networks. In: *IEEE conference on computer vision and pattern recognition*, pp 4584–4593
59. Yu H, He F, Pan Y (2020) A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. *Multimed Tools Appl* 79:5743–5765
60. Zhang J, He F, Chen Y (2020) A new haze removal approach for sky/river alike scenes based on external and internal clues. *Multimed Tools Appl* 79:2085–2107

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Pengjie Tang<sup>1,2</sup> · Jiewu Xia<sup>1,2</sup> · Yunlan Tan<sup>1,2</sup> · Bin Tan<sup>1,2</sup>

Pengjie Tang  
tangpengjie@jgsu.edu.cn

Yunlan Tan  
tanyunlan@163.com

Bin Tan  
tanbin@163.com

<sup>1</sup> College of Electronics & Information Engineering, Jinggangshan University, Ji'an 343009, People's Republic of China

<sup>2</sup> Jiangxi Engineering Laboratory of IoT Technology for Crop Growth, Ji'an 343009, People's Republic of China