ATTENTION-BASED DENSELY CONNECTED LSTM FOR VIDEO CAPTIONING

0) ABSTRACT

- LSTM (Long Short-Term Memory) is commonly used in video captioning due to its ability to handle temporal dependencies.

- However, as sequences get longer, traditional LSTM may struggle with capturing long-range dependencies.

- The research introduces a new model, "Attention-based Densely Connected Long Short-Term Memory" (DenseLSTM).

- In DenseLSTM, all previous cells are connected to the current cell, enabling better information flow.

- An attention mechanism is used to model the impact of different hidden states.

- Each cell in DenseLSTM has direct access to gradients from later cells, improving long-range dependency capture.

- Experiments on MSVD and MSR-VTT video captioning datasets demonstrate the effectiveness of DenseLSTM.

1) INTRODUCTION

- Video captioning is crucial with the growth of the internet and social media.

- Early methods used predefined templates but struggled to capture natural language richness.

- Recurrent Neural Networks (RNNs), particularly LSTM, are effective for video captioning.

- LSTM addresses vanishing/exploding gradients, but long-range dependencies can still be problematic.

- Traditional LSTM needs to pass through multiple cells to use earlier hidden states.

- DenseLSTM is proposed to overcome these issues, maximizing the use of all prior information.

- DenseLSTM directly connects each recurrent cell with all preceding cells.

- It reconstructs temporal information from previous hidden states and feeds it to the current unit.

- Attention mechanisms are used to integrate information from different hidden states.

- DenseLSTM retains long-range information until the last word is predicted.

- The model draws inspiration from DenseNet and utilizes attention mechanisms.

- It realigns previous hidden states based on the global context for video captioning.

- Contributions include introducing DenseLSTM with dense connections, improving sentence accuracy.

- Attention mechanisms help explain the importance of earlier information.

- Evaluation is performed on the MSVD and MSR-VTT datasets.

- Experimental results confirm the effectiveness of the proposed model for video captioning.

2) **RELATED WORK**

2.1 **Video Captioning**

- Early video captioning methods used predefined templates for generating descriptions.

- Template-based methods depend heavily on the quality of templates.

- Recurrent neural networks (RNNs) have become popular for video captioning.

- Researchers use RNNs to model temporal dependencies within video frames and generate sentences.

- Some employ RNNs as decoders, translating visual features into captions.

- Others, like [4, 24, 35], use LSTM as both encoders and decoders for more effective temporal information encoding.

- [4] introduced a boundary detector to segment visual features into chunks, using LSTM for encoding.

- [26] predicts visual attributes in videos and fuses them for latent information exploitation.

- [9] uses topic information, training the encoder and decoder jointly with interpretive loss.

- [7] and [40] exploit visual and audio modalities for encoding and decoding in video captioning.

- [41, 45] incorporate informative regions within video clips by considering temporal structures and object regions.

- Reinforcement Learning (RL) methods like [6, 37] optimize captioning models based on test metrics.

- The focus of this paper is on the decoding stage, following the encoder-decoder framework with DenseLSTM as the decoder for generating target sentences.

2.2 **DenseNet**

- CNNs are widely used for visual tasks, but deep CNNs face vanishing information and gradient issues.

- Recent works introduced shorter connections between inputs and outputs to enable deeper and more efficient training.

- DenseNet, a concept in CNNs, connects each layer to all subsequent layers, promoting feature reuse and addressing gradient vanishing.

- DenseLSTM, similar to DenseNet, introduces shorter connections between LSTM units, addressing the same issues.

- In DenseLSTM, earlier recurrent cell hidden states are fed into successive cells through attention mechanisms, enabling better information flow.

2.3 **Attention Mechanism**

- Attention mechanisms are essential in both visual and natural language processing tasks.

- They help in identifying relevant parts of a source sentence for the target word in neural machine translation.

- Video captioning is similar to translation from input video clips to natural language, and attention mechanisms are used in this context.

- Attention functions map a query and key-value pairs to an output.

- Self-attention focuses on intra-sequence dependencies, where query and key-value pairs come from the same space.

- Self-attention has been used to model dependencies in source sentences and generated target sentences.

- DenseLSTM connects preceding cells with subsequent cells, where information at different times impacts the current word.

- To modulate and integrate different hidden states, DenseLSTM employs self-attention and a task-specific attention mechanism.

3) **METHOD**

3.1 **Long Short-Term Memory**

LSTM is a variant of recurrent neural networks (RNNs) that excels at capturing temporal dependencies in sequential data.

- Given an input sequence {x1, x2, ..., xn }, LSTM generates short-term hidden states {h1, h2, ..., hn } and long-term hidden states {c1, c2, ..., cn }.

- LSTM unit updates for the t-th time step involve several equations:

Input Gate (it):

- Computes it using a sigmoid function.

- Utilizes input xt, previous short-term hidden state ht−1, and long-term hidden state ct−1.

Forget Gate (ft):

- Computes ft using a sigmoid function.

- Utilizes input xt, previous short-term hidden state ht−1, and long-term hidden state ct−1.

Output Gate (ot):

- Computes ot using a sigmoid function.

- Utilizes input xt, previous short-term hidden state ht−1.

Cell Input (дt):

- Computes дt using a hyperbolic tangent function.

- Utilizes input xt, previous short-term hidden state ht−1.

Cell State (ct):

- Updates the cell state ct using it and ft along with the previous cell state ct−1.

Cell Output (ht):

- Updates the short-term hidden state ht using ot and the cell state ct.

- Parameters W∗, U∗, and b∗ are learned during training.

- Note that in this context, "hidden states" refer to short-term hidden states.

3.2 **Baseline Method**

ENCODER:

- UTILIZES A PRE-TRAINED CNN TO EXTRACT VISUAL FEATURES FROM SAMPLED VIDEO FRAMES (V).

- VISUAL FEATURES ARE REPRESENTED AS {F1, F2, ..., FNV}, WHERE FI IS THE FEATURE OF THE I-TH FRAME.

- EMPLOYS A BIDIRECTIONAL LSTM WITH FORWARD (-⟶) AND REVERSE (⟵-) COMPONENTS FOR ENCODING:

  - OUTPUTS FORWARD HIDDEN STATES (-⟶H) AND REVERSE HIDDEN STATES (⟵-H).

- ENCODED FEATURE FOR EACH FRAME IS OBTAINED BY CONCATENATING THE FORWARD AND REVERSE HIDDEN STATES.

- THE ENCODER'S OUTPUTS ARE REPRESENTED AS {H1, H2, ..., HNV}, WHERE HN INCLUDES BOTH FORWARD AND REVERSE COMPONENTS.

DECODER:

- EMPLOYS A TRADITIONAL LSTM AS THE DECODER TO GENERATE CAPTIONS.

- USES HIDDEN STATES (S) FOR DECODING.

- HIDDEN STATE ST AT THE T-TH DECODER CELL IS COMPUTED BASED ON THE PREVIOUS HIDDEN STATE (ST-1) AND THE WEIGHTED SUM OF ENCODER HIDDEN STATES (VT).

- THE ATTENTION MECHANISM IS USED TO COMPUTE WEIGHTS (AT,U) FOR THE ENCODER HIDDEN STATES TO FOCUS ON RELEVANT INFORMATION.

- ATTENTION WEIGHTS ARE TRAINED TO GIVE HIGHER WEIGHTS TO ENCODER HIDDEN STATES THAT MATCH THE CURRENT INPUT DECODER STATE (ST-1).

- ATTENTION FUNCTION (ET,U) IS DEFINED TO COMPUTE THE ATTENTION WEIGHTS.

- PARAMETERS (WA, WS, WH, BA) IN THE ATTENTION LAYER ARE LEARNED DURING TRAINING.

THIS ARCHITECTURE COMBINES AN ENCODER THAT EXTRACTS FEATURES FROM VIDEO FRAMES AND A DECODER THAT GENERATES CAPTIONS USING ATTENTION MECHANISMS FOR IMPROVED ALIGNMENT WITH VISUAL INFORMATION.

3.3 **The Proposed Method**

Overall Architecture:

- Utilizes a bidirectional LSTM as the encoder.

- Employs DenseLSTM as the decoder to generate descriptions.

DenseLSTM Decoder:

- Differs from the traditional LSTM by using short connections between cells.

- Gradients flow directly into preceding units, facilitating training.

Reconstructed Hidden States:

- Instead of directly feeding the last recurrent unit's hidden state (st-1) to the current cell, it reconstructs outputs of all preceding cells {s1, s2, ..., st-1} to form At.

- At is used for decoding.

Two Attention-Based Methods:

- Proposed to effectively fuse {s1, s2, ..., st-1}.

- Overcome the limitation of equal weights in sum or concatenation methods.

- Automatically learn higher weights for more relevant parts during training.

Self-Attention DenseLSTM (Method 1):

- Inspired by self-attention mechanisms used to model dependencies within a sequence.

- Queries (Q), keys (K), and values (V) are the same, where Q = {s0, s1, ..., st-1}, K = {s0, s1, ..., st-1}, and V = {s0, s1, ..., st-1}.

- Scaled dot-product attention is applied to Q and K, followed by softmax to generate weights.

- The weighted sum of values V forms At.

Latest Guiding DenseLSTM (Method 2):

- The input hidden state st-1 is considered an abstraction of previous context.

- Unlike Self-Attention DenseLSTM, where queries and key-value pairs are the same, in Latest Guiding DenseLSTM, the query used for At is st-1, while key-value pairs remain {s1, s2, ..., st-1}.

- At is a weighted sum of decoder hidden states {s1, s2, ..., st-1} with computed attention weights $\beta t,u$.

These two variants provide different methods for reconstructing the hidden state At in DenseLSTM, either through self-attention or guided attention from the latest generated hidden state.

3.4 Sentence Generation

Energy Loss Function:

- The problem involves generating a sentence (W) for a video clip (v) encoded with one-hot vectors.

- The energy loss function is defined as $E(W, v) = -\log(P(W \mid v))$, where $P(W \mid v)$ is the probability of the correct sentence given the input video.

- It represents the negative log probability of the correct sentence.

Log Probability of a Sentence:

- The log probability of a sentence is computed as the summation of the log probabilities over each word, using the chain rule.

- Mathematically, it's expressed as $\log P(W \mid v) = \prod t=1..T \log P(wt \mid v, w0, w1, ..., wt-1)$.

Word Probability Distribution:

- A softmax layer is applied to calculate the probability distribution of the current word (wt) over the word space.

- Probability: $P(wt \mid w0, w1, ..., wt-1, v) \propto \exp(wTt\ Wy\ st)$, where st is the output hidden state from the t-th decoder cell, and Wy represents the parameters of the linear embedding layer.

Optimization:

- The energy loss function is optimized over the entire training dataset.

This formulation allows for the training of the model to generate descriptive sentences for input video clips.

4) EXPERIMENTS

4.1 Datasets

Microsoft Video Description Corpus (MSVD):

- Contains 1,970 short video clips from YouTube.
- Includes 80,839 human-annotated sentences with an average of about 41 sentences per clip.
- Each sentence has approximately 8 words on average.
- Data splits follow those provided in a previous study [35].
- Splits: 1,200 video clips for training, 100 video clips for validation, and 670 clips for testing.

MSR-VTT:

- A dataset for general video captioning, covering a wide range of video categories.
- Comprises 10,000 video clips, each with 20 human-annotated descriptions.
- Adopts the standard split as provided in a previous study [39].
- Splits: 6,513 video clips for training, 497 video clips for validation, and 2,990 video clips for testing.

4.2 Preprocessing and training details

Feature Extraction:

- Static image features and motion features are extracted from input videos using pretrained models.
- VGGNet is used to extract static image features, while C3D is used for motion features.
- Concatenation of these features results in 4096+4096-dimensional visual feature vectors.
- A linear embedding layer is added to the model for input feature vectors.

Data Preprocessing:

- Annotated descriptions are converted to lowercase and tokenized after removing punctuation characters.
- Vocabulary sources vary for different datasets (MSVD uses [35], MSR-VTT retains words appearing at least 5 times).
- Special tokens (<start_token> and <end_token>) are added to captions to represent the beginning and end.

Model Parameters:

- Bidirectional LSTM encoder and decoder share the same parameters.
- Hidden state size for encoder and decoder is set to 1024-dimension.
- Embedding size for input words and visual features is 512-dimension.
- Training learning rate is 0.0001, with parameter initialization in the range [-0.05,0.05].
- Adam optimizer is used for training, with a mini-batch size of 32.
- Dropout with a retain probability of 0.5 is applied to recurrent cells for regularization.

Training Details:

- Video encoder bidirectional LSTM is unrolled to 40-time steps.

- For videos with fewer than 40 frames, padding is applied with zeros.

- For longer videos, only the first 40 frames are retained.

- Language decoder is unrolled to a maximum of 20-time steps.

Evaluation Metrics:

- Four metrics, BLEU, METEOR, ROUGE_L, and CIDEr-D, are used to evaluate the similarity between ground truth and model-generated sentences.

- Microsoft CoCo evaluation toolkit is employed for fair evaluation.

- Results are reported as percentages (%).

This information covers the key aspects of data processing and training for the video captioning model.

4.3 **Experimental Results on MSVD**

Comparisons on MSVD Dataset:

- The study compares the proposed methods (Self-Attention DenseLSTM and Latest Guiding DenseLSTM) with several existing methods, including S2VT, LSTM-E, HRNE, Multimodal Attention, h-RNN, BAE, and MA-LSTM, on the MSVD dataset.

- Evaluation metrics include BLEU@1-4, METEOR, and CIDEr-D, with beam search of size 5.

- The proposed methods outperform the traditional LSTM-based approach as decoders and are comparable to the existing methods.

- Self-Attention DenseLSTM and Latest Guiding DenseLSTM show better performance on CIDEr-D, which is robust to errors in ground truth.

Comparison Between Latest Guiding DenseLSTM and Self-Attention DenseLSTM:

- Latest Guiding DenseLSTM slightly outperforms Self-Attention DenseLSTM on the MSVD dataset. It is suggested that the latest generated hidden states are more important for generating the target word.

Importance of Visual Features:

- The study evaluates the importance of different visual features, such as static image features and motion features. Using both feature types together significantly improves performance compared to using a single modality.

Importance of Attention Mechanisms:

- The study explores the impact of attention mechanisms. Shallow fusion methods (mean-pooling and addition of hidden states) are compared to the proposed methods. More advanced fusion methods, like Latest Guiding DenseLSTM, show superior performance.

Evaluation of Self-Attention DenseLSTM Components:

- Different versions of Self-Attention DenseLSTM are evaluated, with variations in components, including scaled dot-product attention, fully-connected feed-forward networks, and pooling layers. Version 2, which incorporates all components, is found to be the most effective.

Sample Results:

- Sample results are provided, showing that both the baseline method and the proposed methods can generate relevant sentences. The proposed methods generally produce more accurate sentences.

Overall, the proposed methods demonstrate improved performance in generating video descriptions, especially when considering the robustness of evaluation metrics like CIDEr-D. Latest Guiding DenseLSTM is slightly better than Self-Attention DenseLSTM on the MSVD dataset, and combining both static image and motion features enhances overall performance.

4.4 **Experimental Results on MSR-VTT**

Comparison on MSR-VTT Dataset:

- The study compares the proposed models, Latest Guiding DenseLSTM and Self-Attention DenseLSTM, with several existing methods, including MP-LSTM, S2VT, LSTM-E, MA-LSTM, MCNN+MCF, and hLSTMat, on the MSR-VTT dataset.

- Evaluation metrics include BLEU@4, METEOR, and CIDEr-D.

- The proposed models demonstrate better performance compared to the traditional LSTM-based decoder. This result is consistent with the findings on the MSVD dataset.

Utilization of Audio and Category Information:

- The study does not utilize audio and category information provided for videos in MSR-VTT, simplifying the proposed model and reducing the number of parameters. These features have been proven valuable in previous works.

Sample Results:

- Sample generated sentences are provided in Figure 4. The proposed methods generate relevant and coherent sentences.

In summary, Latest Guiding DenseLSTM and Self-Attention DenseLSTM achieve better performance as video captioning decoders on the MSR-VTT dataset compared to traditional LSTM-based methods. The models do not use audio and category information, simplifying the model while maintaining effectiveness.

4.5 **Sentence Generation Analysis**

- The Latest Guiding DenseLSTM computes attention weights for previously generated hidden states.

- The model reveals two important pieces of information:

  1. Typically, the last recurrent cell's information is crucial for predicting the current word.

  2. Some earlier generated hidden states can also be equally or even more important than the last one.

- Figure 5 illustrates the distribution of weights for different hidden states.

- These weights are calculated based on the overall test split of the MSR-VTT dataset.

- To create the weights distribution in Figure 5, weights are initially computed for each video clip, and then the average value for each position is calculated.

- Each column in Figure 5 represents the impact of the hidden state generated at a specific time on predicting a word.

5) **Conclusion**

- The work introduces a model called DenseLSTM to make better use of previously generated hidden states.

- Two attention mechanisms are designed to enhance the fusion of previous hidden states.

- The method computes attention weights, shedding light on the importance of earlier information in predicting the current word.

- Experimental results on two widely used datasets confirm the effectiveness of the proposed approaches.