Enhancing LSTM-based Video Narration Through Text-Derived Linguistic Insights

Abstract

This study delves into how linguistic understanding, extracted from extensive text datasets, can be leveraged to enhance the generation of natural language video descriptions. Specifically, we integrate both a neural language model and distributional semantics, trained on large text corpora, into a contemporary LSTM-based framework for video description. Our evaluation, conducted on a collection of YouTube videos and two substantial movie description datasets, reveals considerable advancements in grammatical correctness, accompanied by subtle improvements in descriptive quality.

1 Introduction

The capacity to automatically generate natural language (NL) descriptions for videos has numerous significant applications, such as content-based video retrieval and aiding visually impaired individuals. Recent effective approaches, use recurrent neural networks (RNNs), treating the problem as a machine translation (MT) task, converting from video to natural language. Deep learning methods like RNNs require extensive training data; however, there's a shortage of high-quality video-sentence pairs. Conversely, vast raw text datasets are readily available, exhibiting rich linguistic structure useful for video description. Most work in statistical MT employs a language model, trained on extensive monolingual target language data, and a translation model, trained on restricted parallel bilingual data. This paper investigates methods to incorporate knowledge from language datasets to capture general linguistic patterns to improve video description.

This study integrates linguistic data into a video-captioning model based on Long Short Term Memory (LSTM) RNNs, known for state-of-the-art performance. Additionally, LSTMs function effectively as language models (LMs). Our initial method (early fusion) involves pre-training the network using plain text prior to training with parallel video-text datasets. Our subsequent two methods, influenced by current MT research, incorporate an LSTM LM with the existing video-to-text model. Furthermore, we explore substituting the standard one-hot word encoding with distributional vectors derived from external datasets.

We present thorough comparisons across these methods, assessing them on a typical YouTube corpus and two recently released extensive movie description datasets. The findings indicate notable gains in description grammaticality (as assessed by crowdsourced human evaluations) and moderate gains in descriptive quality (as determined by human judgements and automated comparisons against human-generated descriptions). Our main contributions include: (1) numerous approaches to integrate knowledge from external text into a current captioning model, (2) comprehensive experiments comparing methods on three large video-caption datasets, and (3) human assessments demonstrating that external linguistic knowledge notably impacts grammar.

2 LSTM-based Video Description

We employ the S2VT video description framework, which we describe briefly here. S2VT adopts a sequence-to-sequence approach that maps an input video frame feature sequence to a fixed-dimension vector, which is then decoded into a sequence of output words.

As depicted in the architecture employs a dual-layered LSTM network. The input to the initial LSTM layer is a sequence of frame features extracted from the second-to-last layer (fc7) of a Convolutional Neural Network (CNN) after the ReLU operation. This LSTM layer encodes the video sequence. At each step, the hidden state is fed into the subsequent LSTM layer. Following the processing of all frames, the second LSTM layer is trained to transform this state into a sequence of words. This can be thought of as using one LSTM to model visual features and another to model language, conditioned on the visual data. We modify this structure to incorporate linguistic information during training and generation. Although our techniques are based on S2VT, they are sufficiently general and could be applied to other CNN-RNN based captioning models.

3 Approach

Current visual captioning models are trained solely on text from the caption datasets and display some linguistic anomalies stemming from a limited language model and vocabulary. Here, we explore several methods to integrate prior linguistic knowledge into a CNN/LSTM network for video-to-text (S2VT) and assess how well they improve overall description quality.

3.1 Early Fusion

Our early fusion method involves initially pre-training the language-modeling components of the network on large raw NL text datasets, before fine-tuning these parameters on video-text paired datasets. An LSTM model can learn the probability of an output sequence given an input. To learn a language model, we train the LSTM layer to predict the next word based on the preceding words. Following the S2VT design, we embed one-hot encoded words into reduced-dimension vectors. The network is trained on extensive text datasets, and its parameters are learned using backpropagation with stochastic gradient descent. The weights from this network initialize the embedding and weights of the LSTM layers in S2VT, which is then trained on video-text data. This trained LM is also utilized as the LSTM LM in both late and deep fusion models.

3.2 Late Fusion

Our late fusion approach draws inspiration from how neural machine translation models incorporate a trained language model during decoding. At each step of sentence generation, the video caption model generates a probability distribution over the vocabulary. We then utilize the language model to re-score the final output by considering a weighted average of the scores from the LM and the S2VT video-description model (VM). Specifically, for output at time step 't', and given proposal distributions from the video captioning model and the language model, we can calculate the re-scored probability of each new word as:

$$p(y_t = y) = \alpha \cdot p_{VM}(y_t = y) + (1 - \alpha) \cdot p_{LM}(y_t = y) \tag{1}$$

The hyper-parameter is tuned on the validation set.

3.3 Deep Fusion

In the deep fusion approach, we integrate the LM more profoundly in the generation process. We achieve this by concatenating the hidden state of the language model LSTM (h_{LM}) with the hidden state of the S2VT video description model (h_{VM}) and use the resulting combined latent vector to predict the output word. This is similar to the method employed to incorporate language models from monolingual data for machine translation. However, our method differs in two ways: (1) We concatenate only the hidden states of the S2VT LSTM and language LSTM, without additional context. (2) We keep the weights of the LSTM language model constant while training the entire video captioning network. The probability of a predicted word at time step t is:

$$p(y_t|G_{<t}, T) \propto exp(W_E(h_t^V \oplus W_T h_t^{LM}) + b) \tag{2}$$

where V is the visual feature input, W represents the weight matrix, and b stands for biases. We avoid fine-tuning the LSTM LM to avoid overwriting previously learned weights of a strong language model. However, the full video caption model is trained to integrate LM outputs while being trained on captioning data.

3.4 Distributional Word Representations

The S2VT network, like many image and video captioning models, uses a one-hot encoding for words. During training, the model learns to embed these one-hot words into a 500-dimensional space via linear transformation. This embedding, however, is learned from the limited and possibly noisy caption data. Many techniques exist that leverage large text datasets to learn vector-space representations of words, capturing nuanced semantic and syntactic structures. We aim to capitalize on these to enhance video description. Specifically, we replace the embedding matrix from one-hot vectors with 300-dimensional GloVe vectors, pre-trained on 6B tokens from Gigaword and Wikipedia 2014. We further explore variations where the model predicts both the one-hot word (softmax loss) and the distributional vector from the LSTM hidden state using Euclidean loss. The output vector (yt) is computed as yt = (Wght + bg), and the loss is:

$$L(y_t, w_{glove}) = ||(W_g h_t + b_g) - w_{glove}||^2 \tag{3}$$

where h_t is the LSTM output, w_{glove} is the GloVe embedding, and W and b are weights and biases. The network becomes a multi-task model with dual loss functions, which we use to influence weight learning.

3.5 Ensembling

The loss function of the video-caption network is non-convex and hard to optimize. In practice, using an ensemble of trained networks can improve performance. We also present results of an ensemble created by averaging predictions from the highest performing models.

4 Experiments

4.1 Datasets

Our language model was trained using sentences from Gigaword, BNC, UkWaC, and Wikipedia. The vocabulary contained the 72,700 most frequent tokens, also including GloVe embeddings. Following evaluation we compare our models on the YouTube dataset, along with two extensive movie description datasets: MPII-MD and M-VAD.

4.2 Evaluation Metrics

We assess performance using machine translation metrics, METEOR and BLEU, to compare model-generated descriptions with human-written descriptions. For movie datasets with a single description, we use only METEOR, as it is more robust.

4.3 Human Evaluation

We also collect human judgments on a random subset of 200 video clips for each dataset through Amazon Turk. Each sentence was evaluated by three workers on a Likert scale from 1 to 5 (higher is better) for relevance and grammar. Grammar evaluations were done without viewing videos. Movie evaluation focused solely on grammar due to copyright.

4.4 YouTube Video Dataset Results

The results show Deep Fusion performed well for both METEOR and BLEU scores. The integration of Glove embeddings considerably increased METEOR, and combining both techniques performed best. Our final model is an ensemble (weighted average) of the Glove model and two Glove+Deep Fusion models trained on external and in-domain COCO sentences. While the state-of-the-art on this dataset is achieved using attention to encode the video our work focuses on language modeling.

Model	METEOR	B-4	Relevance	Grammar
S2VT	29.2	37.0	2.06	3.76
Early Fusion	29.6	37.6	-	-
Late Fusion	29.4	37.2	-	-
Deep Fusion	29.6	39.3	-	-
Glove	30.0	37.0	-	-
Glove+Deep - Web Corpus	30.3	38.1	2.12	4.05*
Glove+Deep - In-Domain	30.3	38.8	2.21*	4.17*
Ensemble	31.4	42.1	2.24*	4.20*
Human	-	-	4.52	4.47

Table 1: Results on the YouTube dataset: METEOR and BLEU@4 scores (in %), along with human ratings (1-5) on relevance and grammar. * denotes a significant improvement over S2VT.

Human ratings align closely with METEOR scores, indicating modest gains in descriptive quality. Linguistic knowledge enhances the grammar of the results. We experimented multiple ways to incorporate word embeddings: (1) GloVe input: Using GloVe vectors at the LSTM input performed best. (2) Fine-tuning: Initializing with GloVe and subsequently fine-tuning reduced validation results by 0.4 METEOR. (3) Input and Predict: Training the LSTM to accept and predict GloVe vectors, as described in Section 3, performed similarly to (1).

4.5 Movie Description Results

Model	MPII-MD		M-VAD	
	METEOR	Grammar	METEOR	Grammar
S2VT	6.5	2.6	6.6	2.2
Early Fusion	6.7	-	6.8	-
Late Fusion	6.5	-	6.7	-
Deep Fusion	6.8	-	6.8	-
Glove	6.7	3.9*	6.7	3.1*
Glove+Deep	6.8	4.1*	6.7	3.3*

Table 2: Results on the Movie Corpora: METEOR (%) and human grammar ratings (1-5). * indicates a significant improvement over S2VT.

The results on the movie datasets show METEOR scores were lower due to single reference translation. Using our architecture, we can see that the capacity of external linguistic information to increase METEOR scores is small yet reliable. Again, human evaluations reveal significant improvements in grammatical accuracy.

5 Related Work

Following the advancements of LSTM-based models in Machine Translation and image captioning, video description works propose CNN-RNN models that create a vector representation of the video, which is decoded by an LSTM sequence model to generate a description. Some works also incorporate external data to improve video description, however, our focus is on integrating external linguistic knowledge for video captioning. We explore the use of distributional semantic embeddings and LSTM-based language models trained on external text datasets.

LSTMs have proven to be effective language models. Other works have developed an LSTM model for machine translation that incorporates a monolingual language model for the target language, achieving improved results. We utilize similar techniques (late fusion, deep fusion) to train an LSTM for video-to-text translation. This model uses large monolingual datasets to enhance RNN-based video description networks. Unlike other approaches where the monolingual LM is used solely for parameter tuning, our approach utilizes the output of the language model as an input for training the full underlying video description network.

Other recent works propose video description models that focus primarily on improving the video representation itself with hierarchical visual pipelines and attention mechanisms. Without the attention mechanism their models achieve good METEOR scores on the YouTube dataset. The interesting aspect is that the contribution of language alone is considerable. Hence, it is important to focus on both aspects to generate better descriptions.

6 Conclusion

This study investigates methods to integrate linguistic knowledge from text datasets for video captioning. Our assessments on YouTube videos and two movie description datasets show improved results according to human evaluations of grammar while also modestly improving the descriptive quality of sentences. Although the proposed methods are assessed on a particular video-captioning network, they are applicable to other video and image captioning models.