Multimodal single sentence video captioning with modality-wise feature reconstruction

Hector Martel
2020280608
Tsinghua University
hkt20@mails.tsinghua.edu.cn

Chua Khang Hui
2020280442
Tsinghua University
ckh20@mails.tsinghua.edu.cn

Abstract

In this document, we explore the task of Video Captioning for our final project of the Natural Language Processing course. This is a research project in which we aim to design and implement a system capable of describing videos with a single English sentence. We approach this problem by considering the video information as a sequence of images plus the audio track, sampled uniformly in 1-second intervals. We propose to combine two existing solutions into a unified framework to enforce reconstruction consistency in our multi-modal setting. We found that incorporating the audio track to the features does help to improve the performance, and that audio-visual reconstruction consistently outperforms the other models in the MSVD and MSR-VTT benchmark datasets.

1 Introduction

With the raise of video platforms that host large collections of videos, describing the content of a scene is crucial to organize the data, perform efficient searches, and extract valuable information from the raw data. Doing it by hand is simply not feasible due to the scale at which these platforms operate and, thus, many approaches have been proposed in recent years to automate this process.

The task of describing a short video using natural language is relatively easy for humans, but presents many interesting challenges to automatic systems. The main reason is the multiple modalities of data involved: visual information, audio track, dialogues and other metadata may be considered together to improve the understanding of the scene. Moreover, the system needs to identify the main entities of the scene, such as humans, objects and events, and how they are related to each other in order to generate a description. The captioning task can be also viewed as a Machine Translation problem between visual and text domains. The entire process usually involves the combination of vision and language models, that share a common latent representation.

The generated text can have variable length and be composed by one sentence (single sentence captioning) or multiple sentences (dense captioning). In this work, we are particularly interested in generating single sentence descriptions. It can be argued that this task is simpler in terms of the amount of text to generate, while it is also substantially harder to obtain a concise and accurate description (even for human annotators). Consequently, the proposed method should exploit the multiple modalities present in the video data by combining Computer Vision, Audio Processing, and Natural Language Processing (NLP) techniques.

2 Related work

The captioning problem is naturally an extension of classification. Instead of providing a single word as a label, the output is a syntactically meaningful sentence. The architecture paradigm that has proven successful to achieve this is the encoder and decoder network. The end-to-end model is composed by a feature extraction step at the encoder level, a feature aggregation step in the latent space and a text generation step at the decoder level (Chen et al., 2019; Islam et al., 2021).

From this point, one sub-problem is the description of a single image (image captioning), to then achieve the description of a sequence of images (storytelling, video captioning). An overview of relevant works in both areas is provided in subsection 2.1 and subsection 2.2, respectively.

2.1 Image captioning

The image captioning problem has been initially tackled as a projection into a triplet space composed by object, action and scene (Farhadi et al., 2010) using statistical methods and sentence templates. With the increasing popularity of Deep Learning, various models used in cascade lead to significant improvements. The intermediate results of image classification and image segmentation models are fed into a text generation model (Fang et al., 2015) to obtain candidate captions, from which the best candidates are selected. Therefore, this method consists of 3 steps: detecting words for entities in the image, generating sentences and re-ranking the sentences.

The first work to propose an end-to-end framework (Vinyals et al., 2015), uses the Inception network (Szegedy et al., 2015a) pre-trained on ImageNet (Deng et al., 2009) as a visual feature extractor followed by an LSTM language decoder. The joint training proved to be very effective and achieved SOTA results, outperforming the baseline methods by a significant margin.

While Show and Tell (Vinyals et al., 2015) relies on extracting global features for the image, DenseCap (Johnson et al., 2016) introduces a Localization Layer after the CNN feature extractor. The motivation is to propose regions of interest in the latent space and generate one sentence for each one of the proposed regions. Therefore, the authors claim that the method can generate dense captions as opposed to the previous method, which generates a single sentence for the global content of the image.

Later, the impressive results of the attention mechanism in other areas motivated its introduction in Show, Attend and Tell (Xu et al., 2016b). This work extends Show and Tell (Vinyals et al., 2015) by using a non-flattened version of the feature maps. This allows the language decoder to apply attention over the relevant parts of the image. The caption is generated word by word, with the decoder having access to the words generated in previous steps.

2.2 Video captioning

As in the early research works from subsection 2.1, video captioning methods existed before the raise of Deep Learning. These methods used hand-crafted features and pre-defined templates, lacking diversity and generalization in the generated text (Guadarrama et al., 2013).

Following the progress of image captioning approaches, the encoder-decoder paradigm can learn more flexible representations and generate richer text outputs. The feature extraction step has been driven by the advances in Computer Vision with 2D and 3D convolutional networks (He et al., 2015; Tran et al., 2015; Carreira and Zisserman, 2018), and convolutional architectures designed for audio like VGGish (Hershey et al., 2017). Feature extraction is usually followed by a feature aggregation step that can be applied over time, modality or space (Venugopalan et al., 2015; Xu et al., 2017; Chen and Jiang, 2019), although some authors claim that feature aggregation remains an open problem in video captioning research (Chen et al., 2019).

RecNet (Wang et al., 2018a) proposes a strategy for video captioning based on reconstructing the visual features from the generated text caption. The objective is to obtain more robust hidden representations of the video content by imposing consistency in the reconstructions, leading to more accurate captions. The authors investigate local and global reconstruction, which are computed frame-by-frame or for the average of the entire video, respectively.

More recently, the outstanding performance achieved by transformer-based architectures has motivated its introduction in the video captioning task. CNNs are still use as a preliminary step to obtain the embeddings that are passed to the transformer network, while the further encoding and the language model happen in the transformer itself (Zhou et al., 2018). An extension of this work (Iashin and Rahtu, 2020), not only uses Bi-SST (Wang et al., 2018b) with 3D convolution (Carreira and Zisserman, 2018) to encode the visual information, but also considers audio embeddings from VGGish (Hershey et al., 2017) and the dialogues obtained from the YouTube captions API [1].

3 Proposed framework

We take inspiration from 2 of the works presented in section 2, namely RecNet (Wang et al., 2018a) and MDVC (Iashin and Rahtu, 2020). Our solution incorporates visual and audio modalities with a feature reconstruction framework. There-

[1] YouTube captions API: https://developers.google.com/youtube/v3/docs/captions

fore, dedicated CNN feature extractors are used for each modality as in MDVC, which are then combined by concatenation and processed by the decoder. The choice for the decoder is a RNN architecture as in RecNet. The captioning architecture diagram is shown in Figure 1.

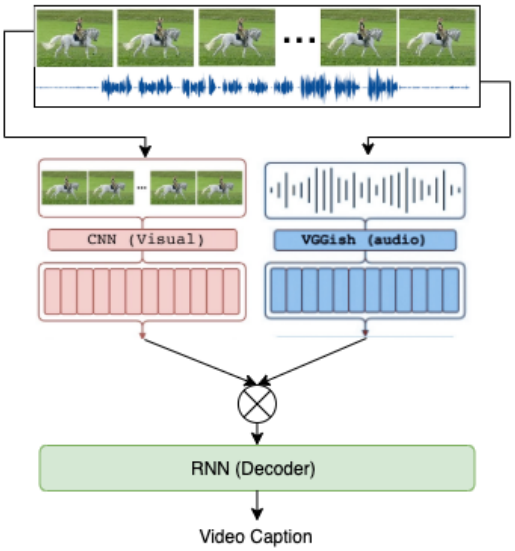


Figure 1: Diagram of the proposed architecture for multimodal captioning. Visual features (red) and audio features (blue) are extracted by 2 different CNNs and then combined before passing them to the RNN decoder.

With this modification, we can reformulate the problem of video captioning for the multimodal setting as follows: Given a sequence of video frames $V \in [0, 255]^{W \times H \times 3}$ and its associated monoaural audio track $A \in [-1.0, 1.0]^{T \times 1}$, our model needs to generate one sentence $S = \{s_1, s_2, ..., s_N\}$ that describes the content in a semantically meaningful way. The goal is to maximize the probability of the output sentence for the given inputs, denoted as $P(S|V, A)$, by tuning the model parameters θ as per Equation 1. Note that the caption is generated word-by-word, and the previously generated words $s_{j<i}$ also condition the next word generation.

$$P(S|V, A) = \prod_{i=1}^{N} P(s_i|s_{j<i}, V, A; \theta) \quad (1)$$

We employ InceptionV3 with ImageNet pretraining together with VGGish as feature extractors, and a LSTM decoder with Soft-Attention (SA) as the captioning module. The feature extractors are detailed in subsection 3.1 and the caption generatior in subsection 3.2.

3.1 Feature extractors

The **visual features** are obtained using the InceptionV3 architecture (He et al., 2015; Szegedy et al., 2015b). This network has been very successful in image classification tasks, and it is a common choice for visual feature extraction. The pre-trained weights from the ImageNet classification challenge (Deng et al., 2009) provide high-level image features due to the diversity of images present in the dataset. This network is a CNN image classifier that works with 2D feature maps through a set of convolutional layers, and then uses a flat representation in 1D to perform a classification with a series of fully connected layers. The output class probabilities are not used directly, but an dense vector from an intermediate fully connected layer is considered instead. We consider a dense vector of 2048 dimensions as frame-level features for the video.

The **audio features** are obtained with the VGGish network (Hershey et al., 2017), which is a CNN that has been re-designed to work with audio data. We use the pre-trained weights from AudioSet (Gemmeke et al., 2017), which can be thought of as an ImageNet equivalent in the audio domain. The embedding provided by this network is a dense feature vector of 128 dimensions at 1-second level audio frames with no overlap.

3.1.1 Feature fusion

One frequent problem in multimodal video captioning is that the data rates for different modalities do not match. In particular, the frame rate of a video can be between 24 and 30 frames-per-second (fps) and the audio sampling rate can vary largely from 16kHz to 48kHz, meaning that the audio contains this number of samples per second.

Other authors (Iashin and Rahtu, 2020) fill the gaps by repeating the information from one modality to match the rate of the other. We take a slightly different approach.

We observe that the default behavior of the VGGish network is to consider equal length audio segments of 1 second. In order to perform the feature fusion by concatenation, we need the visual information to have the same dimensions. Furthermore, the visual and audio data must be consistent in time and keep their original alignment. Therefore, we downsample the video at 1 fps to match the audio features.

Our assumption is that a pair of frames that oc-

cur in less than 1 second will result in very close high-level image features, which would be redundant information. This downsampling feature fusion strategy allows us to achieve the following 2 things: 1) the data modalities are sampled using the same time intervals, thus, the obtained features match despite the differences in the original data rates, and 2) we significantly reduce the length of the sequences that are passed to the decoder by removing *near-duplicate* features, which main benefit is the reduction of computational demands during training.

3.2 Captioning module

As previously mentioned in the introduction of section 3, the output captions are generated word-by-word, using the fusion of features and the previous words in the caption as context information. The token sequences are passed to a stack of LSTM layers to obtain a hidden representation and the next word. During training, the ground truth words are used as context input for the next word prediction, which is a common practice for sequence learning called *teacher forcing*. In particular, the amount of contribution from the ground truth caption and the generated words is controlled by a value between 0 and 1.

A Soft-Attention (SA) mechanism is applied over the time dimension of the video and audio features to select a subset of 1-second segments that are the most relevant to generate the captions.

The architecture of the captioning module is shown in Figure 2.

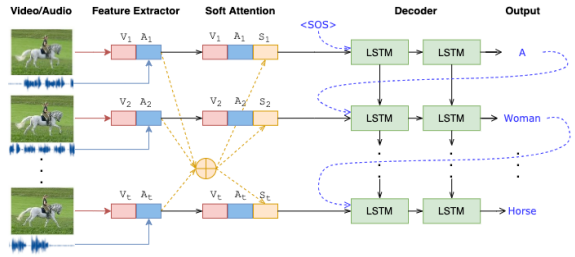


Figure 2: Architecture diagram of the captioning module (SA-LSTM decoder). Visual features (red) and audio features (blue) are concatenated after extraction. The Soft-Attention mechanism assigns attention weights (yellow) that are also passed to the LSTM layers for sentence generation.

3.3 Reconstruction module

Following the strategy suggested in RecNet (Wang et al., 2018a), we impose a reconstruction consistency on the features and the generated captions, with the aim to obtain a more robust latent representation in the decoder. We extend the idea from the original paper to separately reconstruct the features of individual modalities, namely visual and audio, from the caption generated by the model.

The extension for both data modalities allows us to design a more flexible control in the loss function. Following the terminology from Equation 1, let $S = \{s_1, s_2, ..., s_N\}$ denote the output caption, V the visual features, $\hat{V}$ the reconstructed video features, A the audio features, and $\hat{A}$ the reconstructed audio features. The coefficients λ_{visual} and λ_{audio} can be adjusted to change the amount of contribution of each modality. L_{rec} represents the reconstruction loss, which can be *local* or *global*. The total optimization objective is expressed in Equation 2, where θ_{visual_rec} and θ_{audio_rec} denote the parameters for the modality reconstructors.

$$\begin{aligned} &\sum_{i=1}^{N}(-logP(s_i|s_{j<i}, V, A; \theta)) \\ &+\lambda_{visual} \times L_{rec}(V, \hat{V}, \theta_{visual\_rec}) \\ &+\lambda_{audio} \times L_{rec}(A, \hat{A}, \theta_{audio\_rec})) \end{aligned} \tag{2}$$

For **local** reconstruction, the objective is to obtain a sequence of feature vectors that approximates the original input features. The number of feature vectors to reconstruct is N, which corresponds to the length of the clip in seconds for our extracted features. In other words, let the local visual features $\{V_1, V_2, ..., V_N\}$ and their reconstructions $\{\hat{V_1}, \hat{V_2}, ..., \hat{V_N}\}$ and the local audio features $\{A_1, A_2, ..., A_N\}$ and their reconstructions $\{\hat{A_1}, \hat{A_2}, ..., \hat{A_N}\}$. The local reconstruction loss is calculated as the Mean Squared Error (MSE) between frame-wise pairs of original and reconstructed feature vectors, i.e. $MSE(V_i, \hat{V_i})$ and $MSE(VA_i, \hat{A_i})$ for $i \in [1, N]$.

For **global** reconstruction, the objective is to recover a *context vector* of the features that is obtained from the entire clip. This vector is computed with mean pulling as the mean along the time dimension, i.e. V_{mean} and $\hat{V}_{mean}$ for visual features, and A_{mean} and $\hat{A}_{mean}$ for audio features. The global reconstruction loss is the given by the MSE between the true features and the reconstructed features: $MSE(V_{mean}, \hat{V}_{mean})$ and $MSE(A_{mean}, \hat{A}_{mean})$.

After computing the appropriate reconstruction loss, the information is back-propagated to the

captioning module to update the free-level representations. A graphical illustration of the reconstruction model is shown in Figure 3 and the *local* and *global* reconstruction losses are shown in Figure 4.

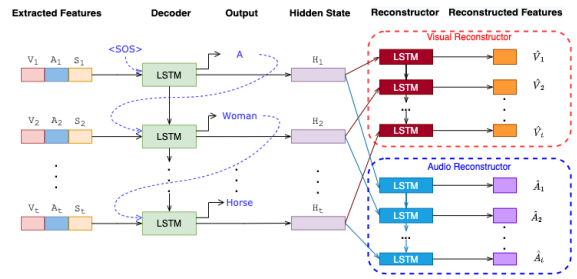


Figure 3: Architecture diagram of the Reconstructor, with Video (red) and Audio (blue) modalities in separate modules. The reconstructor receives the hidden state of the LSTM decoder E_i and pass it through another LSTM to obtain the reconstructed sequence of features for video ($\hat{V}_i$) and audio ($\hat{A}_i$).

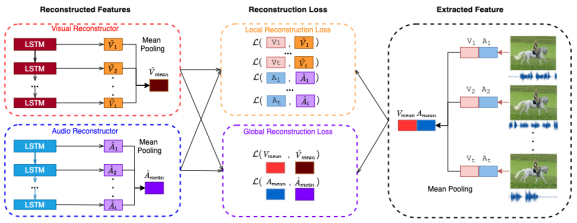


Figure 4: Illustration of the Reconstruction loss. The features $(\hat{V}_i, \hat{A}_i)$ from the modality reconstructors (on the left) are compared with the original features $((\hat{V}_i, \hat{A}_i)$, on the right) to compute the corresponding loss, global or local.

3.4 Model training

The training is performed in a supervised setting. The feature extractors are not trainable, as they are used with ImageNet and AudioSet pre-training. Only the decoder and the reconstructors are optimized. The encoder receives the pre-computed features extracted by the CNNs as input. The ground truth captions are provided as output and, depending on the *teacher forcing* parameter, previous words from the ground truth captions are used in the sentence generation process. In the base LSTM decoder, we have found that larger values of teacher forcing benefit the convergence to better solutions. We experimented with 0, 0.2 0.5 and 1.0, and finally set it to 1.0 due to its superior performance.

All the models are trained using Adam optimizer, with a learning rate of 2×10^{-4} and a batch size of 128. The rest of the optimizer hyperparameters are left as default. To prevent the model converging to a local minimum, the learning rate is divided by a factor of 2 if the validation loss does not improve for 5 consecutive epochs, with a lower bound value of 1×10^{-7}. The number of epochs is set to 50 for MSVD and 30 for MSR-VTT (see datasets in subsection 4.1).

The values are fixed for all the experiments in order to establish a fair comparison between all the models. The reconstruction loss coefficients λ_{visual} and λ_{audio} are set to 5×10^{-1} and 5×10^{-5}. These values were found to provide a good compromise between the loss terms after several runs with different parameter configurations. Finally, an entropy regularization term is set to 5×10^{-4}.

4 Experiments

This section presents the data and evaluation used in our experimental setting. An overview of the datasets and data processing steps is given in subsection 4.1. The evaluation metrics are introduced in subsection 4.2. Finally, objective and subjective results are presented in subsection 4.3.

4.1 Datasets

Microsoft Research Video Description Corpus (MSVD) (Chen and Dolan, 2011) is one of the common open domain datasets for the video captioning task. MSVD consist of 1970 video snippets extracted from YouTube with around 40 English annotated captions for each snippet. In older video captioning literature, MSVD is also referred to as the YouTube2Text dataset (Guadarrama et al., 2013). The duration of each video is typically between 10 to 25 seconds, mainly showing one activity per clip.The standard split of this dataset is 1200 for training, 100 for validation and 670 for testing.

The original MSVD dataset is provided *without audio tracks*. Therefore, as an additional data processing step, we download the videos from their source URLs on YouTube and extract the audio tracks for our multi-modal setting. At the time of writing, only 75% of the videos were available for download.

Furthermore, we apply balancing on the dataset captions to remove duplicated entries, as we observed that the models were highly biased towards generating only a reduced set of sentences. A caption is considered to be a duplicate if the same pair

of $(video\_id, caption)$ appears more than once. This reduces the number of captions per clip to about 20, instead of 40.

Our train, validation and test splits considering audio tracks and without duplicates result in 624, 60 and 392 videos, respectively. This is roughly a 50% of the original dataset splits.

Microsoft Research Video to Text (MSR-VTT) (Xu et al., 2016a) is a large-scale video benchmark dataset that contains 7180 videos subdivided into 10,000 short clips. The duration ranges between 10 and 30 seconds, and the clips cover a wide variety of categories and actions. Each clip has about 20 natural sentence descriptions annotated by Amazon Mechanical Turk (AMT) workers. The standard split for this dataset is composed of 6513 videos for training, 497 videos for validation, and 2990 videos for testing. MSR-VTT is one of the largest video captioning datasets and it it provides visual and audio information. Thus is more suitable than MSVD for our multi-modal setting, as no further data preparation steps are required.

4.2 Evaluation metrics

Since this is a sequence-to-sequence problem, one approach for objective evaluation is to use Machine Translation (MT) evaluation metrics for this problem as well. Common evaluation metrics for video captioning task include BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and METEOR (Lavie and Denkowski, 2009). The results are evaluated based on the semantic relevance of the of generated caption S_{gen} with respect to the annotated ground truth caption S_{gt}.

BLEU measures the effective overlap between S_{gen} and S_{gt}. BLEU@n is calculated by the multiplication of geometric mean of the n-gram precision score of the matching words. It is sensitive to position mismatching of words, hence it tends to favor shorter sentences. Therefore, a brevity penalty factor is introduced to penalize the short translations. Our motivation to use this metric is that, in this single-sentence generation setting, the captions are relatively short in both datasets.

ROUGE measures the overlapped sub-sequences of tokens between S_{gen} and S_{gt}. This measure highly depends on recall, so it favors longer sentences. ROUGE-L specifically focus on the *longest* common sub-sequence of two sentences. BLEU and ROUGE are effective in measure the similarity between reference and candidate sentence, however, both metrics are weakly correlated with human judgment.

CIDEr was originally designed for image captioning evaluation, and it can be extended to video captioning. It measures the similarity of a generated sentence S_{gen} against a set of human-annotated ground-truth sentences $\{S_{gt}^1, S_{gt}^2, ..., S_{gt}^N\}$ by applying the consensus between the annotators. This means that the score is highly correlated to human judgements. However, this implies that the score depends on the number of reference annotations and their quality, which can be problematic when they are scarce.

METEOR measures the distance of the semantic meaning of unigrams from a reference sentence to a set of candidate sentences. It compares the exact tokens, stemmed tokens, and synonyms, to better measure the exact and paraphrase matching between S_{gen} and S_{gt}. For this reason, METEOR gives a more accurate evaluation than CIDEr when the number for references is small.

In summary, BLEU and ROUGE are good at measuring the differences in exact word matches between generated sentence and ground-truth sentences depending on their lengths, whereas CIDEr and METEOR are better at measuring the semantic differences because they incorporate knowledge for annotation consensus or paraphrasing. In this project we consider BLEU@4, ROUGE-L, METEOR and CIDer for result evaluation.

The evaluation metrics are calculated using the Python package `pycocoevalcap`[2].

4.3 Experimental results

Here we present the results of our experiments according to the evaluation metrics presented in subsection 4.2, and we attach some concrete video examples. Objective results can be found in Table 1 for MSVD and Table 2 for MSR-VTT. Additionally, some captioning examples are included in Figure 5 (at the end of this document).

We perform an ablation study by trying different configurations of input modalities and reconstructions. The values reported with V correspond to *video only*, while $A + V$ corresponds to *audio and video*. To train the models in *video only* mode, we set the input audio features to a vector of zeros, effectively suppressing the audio track. For Ta-

[2] `github.com/salaniz/pycocoevalcap`

ble 1 and Table 2, the baseline models correspond to those denoted with V, and the proposed models to $A + V$. For simplicity, and to avoid all combinations, we set the same reconstruction strategy to either *global* or *local* for both modalities. For sentence generation, the maximum length is set to 30 tokens with direct predictions. Beam search with a width of 5 was considered in early experiments, but has been discarded due to its higher computational cost and the lack of improvement in the results.

In the examples from Figure 5, we present 4 videos from either the validation or test splits of the MSR-VTT dataset, with various subjects and actions. Regardless of the original duration of the clip, the figures display 5 frames sampled uniformly along the time dimension to represent the clip. Below each clip, there are 3 types of captions. *Baseline model* refers to the SA-LSTM trained with only visual information (V). *Best model* refers to the best performing model on the dataset, which according to Table 2, is the $A + V$ decoder with $A + V, Global$ reconstruction. Finally, 2 ground truth reference sentences are provided for comparison. The words in all captions are color-coded to illustrate to what extent the predictions are correct. See Figure 5 for details.

5 Conclusions

In this work, we have presented a system to generate single-sentence captions given video data as an input. We have studied the effectiveness of using multi-modal data from the video, considering a joint representation of the visual and audio information.

The following discussion is intended to cover objective evaluations, some comments based on the manual inspection of the output, and other challenges that remain an open research question.

5.1 Objective evaluation

From our experiments, it can be concluded that incorporating multi-modal information is clearly beneficial to the models' performance. Moreover, the performance gaps between V and $A+V$ models are consistent in the two datasets. This experimental outcome is expected, as models with more available information can take advantage of it to generate better output captions. Nevertheless, we have also observed that the training process in the multi-modal setting presents more instability and takes longer to converge. The reason is that the visual and audio of some videos from the datasets are not well correlated, such as in the presence of background music, high levels of noise, or occluded speakers.

The following analysis prioritizes the results on the MSR-VTT dataset, since it is larger and its annotations are of better quality than MSVD (see subsection 5.3 for more detailed discussion).

In MSR-VTT (see Table 2), all the BLEU@4 and CIDEr scores obtained by $A + V$ models are superior to V by a large margin. In MSVD (see Table 1), the same holds true for the CIDEr scores. ROUGE_L and METEOR do not seem to show significant differences between the models in either dataset.

Let us focus on the evaluation results of $A + V$ modalities. They include various feature reconstruction strategies, *local* and *global* for V and $A + V$ modalities, namely $V, Local$, $V, Global$, $A + V, Local$ and $A + V, Global$. When both modalities are considered in the reconstruction, the scores are slightly higher than models with only visual reconstructor or without any reconstructor.

Given our current data, we cannot make any comments on which strategy is better, as it is dependent on the dataset. In MSR-VTT, the $A + V, Global$ provides the best performance, whereas in MSVD, it is the $A + V, Local$ and $V, Global$ that perform the best. Therefore, despite the apparent success of the reconstruction strategies, it is hard to tell with high levels of confidence whether or not they benefit the models in our multi-modal setting, and what is the best choice.

The intuitive answer is that the reconstruction in both modalities does not degrade the performance and, indeed, it is helpful to learn robust feature embeddings in the decoder. Furthermore, the $Global$ reconstruction seems to convert the reconstruction in a more tractable problem, as only the context, and not individual feature vectors, are used to compute the loss. In this sense, a good global reconstruction is an indicator that the caption is a complete summary of the video, rather than a partial summary. Ultimately, this correlates better with the objectives of the video captioning task.

Decoder	Modalities	Reconstructor	Bleu_4	METEOR	ROUGE_L	CIDEr
SA-LSTM	V	-	0.223	0.223	0.625	0.350
SA-LSTM	V	V, Local	0.207	0.223	0.627	0.246
SA-LSTM	V	V, Global	0.204	0.230	0.631	0.266
SA-LSTM	A+V	-	0.239	0.251	0.655	0.408
SA-LSTM	A+V	V, Local	0.217	0.238	0.624	0.383
SA-LSTM	A+V	V, Global	0.237	**0.254**	0.647	**0.485**
SA-LSTM	A+V	A+V, Local	**0.268**	0.249	**0.661**	0.406
SA-LSTM	A+V	A+V, Global	0.246	0.243	0.652	0.384

Table 1: Experiment results on the MSVD dataset. For all the evaluation metrics, higher values indicate better performance.

Decoder	Modalities	Reconstructor	Bleu_4	METEOR	ROUGE_L	CIDEr
SA-LSTM	V	-	0.273	0.221	0.532	0.167
SA-LSTM	V	V, Local	0.281	0.226	0.533	0.189
SA-LSTM	V	V, Global	0.255	0.248	0.523	0.127
SA-LSTM	A+V	-	0.343	0.248	**0.568**	**0.293**
SA-LSTM	A+V	V, Local	0.338	0.243	0.561	0.288
SA-LSTM	A+V	V, Global	0.345	0.247	0.562	0.279
SA-LSTM	A+V	A+V, Local	0.353	**0.249**	0.565	0.285
SA-LSTM	A+V	A+V, Global	**0.356**	**0.249**	**0.568**	0.290

Table 2: Experiment results on the MSR-VTT dataset. For all the evaluation metrics, higher values indicate better performance.

5.2 Subjective evaluation

On top of looking at the evaluation scores, our ultimate goal is to verify that the system can generate outputs that are coherent for human subjects. For this reason, we want to discuss some of the insights that we have extracted from Figure 5.

In general, the sentences are related to the ground truth captions in terms of overall meaning. The precise details from the ground truth sentences are not always captured by the caption generation models. We have observed that the length of the generated captions tends to be shorter or equal to the ground truth captions, but rarely exceed it.

From the **first clip**, we can extract that the occlusion of the speakers (either the man that is partially outside of the scene at the end, or the other man holding the newspaper covering his face) plays an important role in determining the genres of the characters. While the baseline model predicts a woman based solely on visual information, the audio information can be used by the best model to determine that the subject is a man, supported by the speech audio.

The **second clip** illustrates the limitations of the current multi-modal approach. Human annotations reflect that the woman in the news is talking about a certain topic (politics, elections), but this is not being captured by the models. Instead, the output is a more generic (news, story). This can be overcome by using a Automatic Speech Recognition (ASR) as another feature extractor to supply the dialogues as side information to the captioning model.

The **third clip** shows another example of the audio being important in the caption generation, as well as some model biases. While the baseline model predicts that the subject is a *man* and that he is *driving* the car, in reality, it is not clear from the visual information alone. In this case, subject and its gender are predicted correctly because this choice is likely in the data, but there is not enough information from the scene. Following the same logic, the baseline model predicts the action of *driving* because of the presence of a car in the clip, but the car is static. The best model can generate a more accurate caption by considering that there is a male speaker's voice in the background and no engine noise.

Finally, the **last clip** is predicted correctly in the most part. However, we can see that the ground

truth annotations are not describing the exact same actions. The first one includes the verb *to explain* (involves talking, but maybe not doing the actions while speaking) whereas the second one uses *to prepare* (does not necessarily mean to talk, but to do the actions). We would expect the best model to output a sentence with the verbs *to talk*, *to explain*, or similar if the first case was true.

From these comments, we think that there is still a lot of room for improvement in this captioning models before the results can be comparable to human-level performance in any meaningful way.

5.3 Dataset limitations

Another important observations are concerning the MSVD dataset. MSVD is used in many research papers, which is one of the main reasons we decided to include it in our experiments. Nevertheless, we are skeptical about whether its results are truly representative of the video captioning task. We would like to highlight two important issues that we have found in the MSVD dataset during our experiments. We hope these insights are useful to re-think the benchmarks of the video captioning task.

Poor annotation quality. Manual inspection of the ground truth annotations has revealed that a significant portion of them are inconsistent, lack diversity, or do not explain the video contents precisely enough. To illustrate all these issues in one example, consider the sentence *"Someone is doing something"*, which is present in the annotated data. This caption not only can be applied to any video with a person on the scene, but it reduces the effectiveness of the training process. We believe that, while it is possible to obtain high evaluation scores by training a model on such data, its generalization capability in a real case would be extremely low.

Available data is not enough. A column is provided in the metadata to indicate if sentences have been verified or not. Verified sentences are assumed to be more representative of the video contents than the unverified ones. As mentioned in subsection 4.1, we needed to clean the data in order for the models to be able to learn on MSVD. The steps of data cleaning included selecting only verified annotations, removing duplicates, and removing videos without available audio tracks. In our views, even though the data is in an adequate format, the number of remaining examples is simply not sufficient to train a deep learning model that can generalize to unseen data.

5.4 Future work

It would be useful to do a more careful search of the optimal hyperparameters. The current results have been obtained in a fixed experimental setting, which we argue is still suboptimal. Our results can be compared between the models that we have trained, but not with the current state of the art. We would like to mention some points for future work:

- In terms of feature extraction, we have used a frame down sampling at 1 fps to achieve alignment with the audio track and reduce the computation. However, this is a possible cause of the degradation in performance when compared to other methods that use frame rates between 5 fps and 25 fps, and compute the audio in overlapping windows.
- For the loss function, the coefficients λ_{visual} and λ_{audio} may be further tuned to improve the performance of $A + V$ reconstructions.
- For the model architecture, the capacity of the SA-LSTM decoder can be tuned by adjusting the number of layers, dropout rate, and attention bottleneck. In addition, another attention mechanism can be incorporated for modality feature fusion in replacement of the simple concatenation. Time and modality-wise attention combined should provide a boost in performance to our existing solution. Alternatively, more advanced architectures based on multi-modal transformers can be explored, although considering that the requirements needed to train such models are significantly higher.

References

Joao Carreira and Andrew Zisserman. 2018. Quo vadis, action recognition? a new model and the kinetics dataset.

David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA. Association for Computational Linguistics.

Shaoxiang Chen and Yu-Gang Jiang. 2019. Motion guided spatial attention for video captioning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8191–8198.

Shaoxiang Chen, Ting Yao, and Yu-Gang Jiang. 2019. Deep learning for video captioning: A review. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6283–6290. International Joint Conferences on Artificial Intelligence Organization.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, page 15–29, Berlin, Heidelberg. Springer-Verlag.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *2013 IEEE International Conference on Computer Vision*, pages 2712–2719.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. 2017. Cnn architectures for large-scale audio classification.

Vladimir Iashin and Esa Rahtu. 2020. Multi-modal dense video captioning.

Saiful Islam, Aurpan Dash, Ashek Seum, Amir Raj, Tonmoy Hossain, and Faisal Shah. 2021. Exploring video captioning techniques: A comprehensive survey on deep learning methods. *SN Computer Science*, 2.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2–3):105–115.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015a. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015b. Rethinking the inception architecture for computer vision.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence – video to text.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator.

Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. 2018a. Reconstruction network for video captioning.

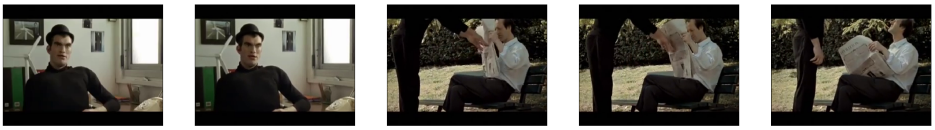
Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018b. Bidirectional attentive fusion with context gating for dense video captioning.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016a. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. 2017. Learning multimodal attention lstm networks for video captioning. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 537–545, New York, NY, USA. Association for Computing Machinery.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2016b. Show, attend and tell: Neural image caption generation with visual attention.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer.



Baseline model: a man is talking to a woman

Best model: a man is talking to a man

Ground truth: a man is talking to another man seated reading the newspaper
a man approaches a man on a bench

Baseline model: a woman is talking about a news

Best model: a woman is talking about a news story

Ground truth: a female news anchor discussing politics
a woman is talking about election in us

Baseline model: a man is driving a car

Best model: a man is talking about a car

Ground truth: a man outside talking about a car
a man shows off a new car and talks about the features

Baseline model: a man is cooking food

Best model: a man is cooking a dish in a kitchen

Ground truth: a chef explains how to cook a recipe
a man is preparing a meal in a kitchen

Figure 5: Examples of video captioning results from the MSR-VTT dataset. The generated captions from the baseline SA-LSTM (V, no reconstructor) decoder and the best performing model SA-LSTM ($A + V, Global$) are presented together with 2 ground truth captions. Green: correct words comparing the generated captions with the ground truth captions; Orange: partially correct or ambiguous words; Red: wrong words in the generated captions.