Transforming Videos to Text (VTT Task)
Team: MMCUniAugsburg

Philipp Harzig
Multimedia Computing and Computer Vision Lab
University of Augsburg
Augsburg, Germany
philipp.harzig@uni-a.de

Moritz Einfalt
Multimedia Computing and Computer Vision Lab
University of Augsburg
Augsburg, Germany
moritz.einfalt@uni-a.de

Katja Ludwig
Multimedia Computing and Computer Vision Lab
University of Augsburg
Augsburg, Germany
katja.ludwig@uni-a.de

Rainer Lienhart
Multimedia Computing and Computer Vision Lab
University of Augsburg
Augsburg, Germany
rainer.lienhart@uni-a.de

***Abstract*—The Multimedia and computer Vision Lab of the University of Augsburg participated in the VTT task only.**
We use the Auto-captions on GIF [1] (AC-GIF), MSR-VTT [2] and TRECVID-VTT [3] datasets for training our VTT models.
We base our model on the Transformer [4] approach for both of our submitted runs, i.e., for runs *103.primary* and *102*. For our *103.primary* run, we use the complete MSR-VTT dataset and 90% of the TRECVID-VTT dataset for pretraining while using the remaining 10% for validation. For the *102* run, we additionally utilize the complete AC-GIF dataset for the pretraining stage. Both runs were finetuned on TRECVID-VTT (90%). During finetuning the *102* run, the validation performance decreases significantly, while *103.primary* improves in performance. The use of the AC-GIF dataset decreases the performance, because the domain and the captions are different to the other datasets.
Overall, we find that training a Video-to-Text system on traditional Image Captioning pipelines [5] delivers very poor performance. When switching to a Transformer-based architecture our results greatly improve and the generated captions match better with the corresponding video (see Figure 3).

I. Introduction

In this notebook paper, we present our Video-to-Text model, which allows to create descriptions for arbitrary videos. Our model is inspired by the classical Transformer [4] approach.

II. Model

A. Preprocessing of Videos

In order to process the videos in our model, we first need to extract single frames. We use ffmpeg for extracting every frame of each video of the respective dataset. We use ResNet-101 [6] to compute features for the extracted frames. More specifically, we resize the input images to 224×224 and use the average pooled features with dimension $\mathbb{R}^{2048}$.

B. Model

An overview of our model architecture is depicted in Figure 1. In comparison with the original Transformer [4] architecture, we changed the encoder part to accept image features instead of embedded words. That is, we exchanged the sentence encoder with a video encoder. More specifically, we replaced the input embedding with an image embedding, which is standard practice in common image captioning models [5]. An image embedding layer embeds the image features into the desired embedding space. In our model, we use ResNet-101 features $\in \mathbb{R}^{2048}$ and embed them into the encoder space with dimension $d_{\text{model}} = 512$.

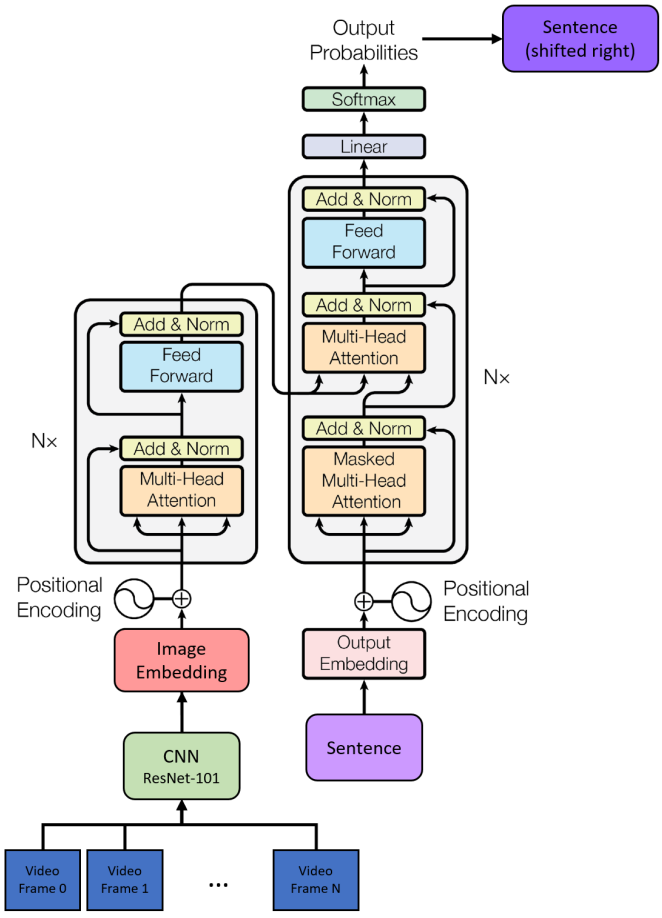

Fig. 1. Our model architecture was slightly modified from the original Transformer [4] to allow video frames as input to the encoder blocks. Original image taken from [4] and modified to match our architecture.

We also use positional encoding to encode the order of every single frames in the video. As the Transformer architecture does not care about the order of the input, i.e., every frame can influence every

TABLE I

THE THREE DATASETS WE USED TO TRAIN OUR MODELS. WE LIST THE NUMBER OF VIDEOS AND SENTENCES IN THE DATASET AND THE NUMBER OF USABLE VIDEOS AND SENTENCES, I.E., VIDEOS THAT WERE AVAILABLE AT THE TIME OF DOWNLOAD.

Dataset	# Videos (clips)	# Sentences	# Videos used	# Sentences used
AC-GIF [1]	163,183	164,378	163,183	164,378
TRECVID-VTT [3]	7,485	28,183	5,971	22,547
MSR-VTT [2]	10,000	200,000	7,773	155,460

other frame in the same way, we need to explicitly tell the encoder the frame number. Similar to the original paper, we use a positional encoding to encode the frame number, which we add on top of the embedded image features.

The rest of the Transformer architecture is nearly identical to the proposed architecture in [4]. In the encoder, we made use of the memory-augmented encoding [7], which encodes multi-level visual relationships with a priori knowledge. In the original work, Cornia et al. use a persistent, learnable memory vector which is concatenated to the key and value of the self-attention blocks of the Transformer (see Figure 2). These memory vectors allow to encode persistent a-priori knowledge about relationships between image regions. In contrast to the original work, we work with video sequences instead of still images with regions. Adapted to our architecture, the memory vector encodes a-priori knowledge about relationships between frames in a given video. We did not change the architecture of the decoder block.

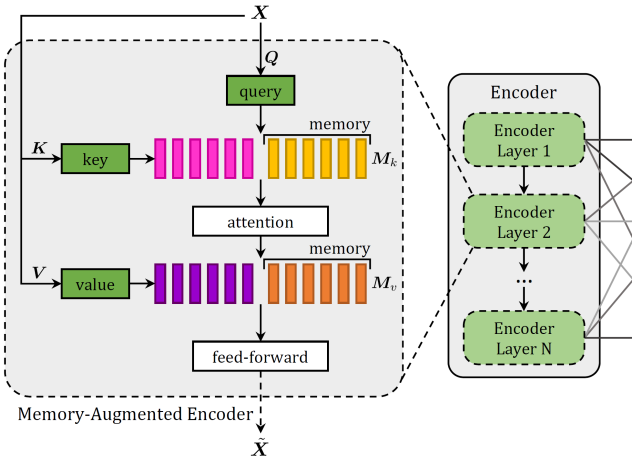

Fig. 2. The Memory-Augmented Encoder, which we used in our Multi-Head-Attention blocks. Image taken from [7].

III. DATASOURCES

We use three datasets for training our models, which are described below. Additionally, we show some dataset statistics in Table I.

A. Auto-captions on GIF

The *Auto-captions on GIF* [1] (AC-GIF) dataset was designed for pre-training Video-to-Text models. Because existing video-sentence datasets are mostly task-specific, i.e., they are mainly focused on specific domains such as cooking [8], the authors of the AC-GIF dataset tried to create a more generic dataset. They created their dataset by collecting GIFs and their respective alt-text HTML attributes from the web. The AC-GIF dataset contains $163,183$ videos and $164,378$ sentences. The total number of words is $1,619,648$ with an vocabulary of $31,662$ words.

TABLE II

DATA SOURCES USED FOR TRAINING OUR BASE MODELS. WE ALSO DEPICT THE TOTAL NUMBER OF TRAINING AND VALIDATION SAMPLES USED.

Model: Data sources	# train samples	# val samples
1: MSR-VTT + 90% TRECVID-VTT	175902	2285
2: MSR-VTT + AC-GIF + 90% TRECVID-VTT	303380	2285

B. TRECVID-VTT

We use the official TRECVID-VTT dataset [3] which contains videos from the TRECVID VTT from 2016-2019. We only use the Twitter Vine subset of videos. In total, this subset contains $6,475$ videos from which we use $5,971$ available videos with $22,547$ captions. In all our experiments we train on 90% and validate the model on 10% of the videos.

C. MSR-VTT

We also use the MSR-VTT dataset [2] for training our VTT model. The MSR-VTT dataset consists of $7,180$ videos, which make up $10,000$ clips. In total, the dataset contains $200,000$ sentences with a total of $1,856,523$ words and a vocabulary of $29,316$ words. Because not all videos were available at the moment of download, we only use $7,773$ video clips with $155,460$ corresponding sentences.

IV. MODEL CONFIGURATIONS

We submitted two models for the Video-to-Text (VTT) task. Both of our models are pretrained on a merged dataset and then finetuned on the TRECVID-VTT dataset. For our primary model (cf. *103.primary*), we first train a base model on the full MSR-VTT dataset and 90% of the TRECVID-VTT dataset. We select the model by employing an early-stopping strategy on the CIDEr score of the remaining 10% of the TRECVID-VTT dataset. For finetuning, we use the base model and train it on the 90% split of the TRECVID-VTT dataset. We also use early-stopping to select our final primary model.

Our second model (cf. *102*) is trained similarly, except we use AC-GIF (full), MSR-VTT (full) and TRECVID-VTT (90%) for training the base model. For finetuning the second model, we also use the TRECVID-VTT (90%) split. In Table II, we present the number of training samples used for training the base and finetuned models.

Our models use 8 encoder and 8 decoder blocks. We use 8 attention heads and a model dimension of $d_{\text{model}} = 512$. For the position-wise feed-forward networks, we set $d_{ff} = 2048$ as the inner-layer dimensionality. We use a memory-vector size of $d_{\text{memory}} = 64$. The primary model uses a vocabulary of $12,000$ complete words. For the second model, we use a subword text encoder with $20,283$ subwords. It does not use complete words for the vocabulary, but tries to build words from subwords, i.e., it splits words into subwords if a word is not in the initial dictionary.

V. TRAINING

We train our models in a multi GPU setting, i.e., we train the model on 5 NVIDIA Tesla V100 GPUs simultaneously. We use a batch size of 16 per GPU, resulting in an effective batch size of 80. We use the Adam [9] optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$. Similar to [4], we used a variable learning rate η over the course of the training. That is, we used a linearly increasing learning rate

TABLE III
SUBMITTED MODELS (IN BOLD) AND THEIR RESPECTIVE VALIDATION SCORES. WE VALIDATED ALL OF OUR MODELS AFTER EVERY EPOCH ON 10% OF THE TRECVID-VTT DATASET TO SELECT A MODEL TO SUBMIT.

Model	best performance @ epoch	B-1	B-2	B-3	B-4	CIDEr	METEOR
primary-base	25	0.4104	0.2340	0.1323	**0.0760**	0.1755	0.1161
primary-ft	8	**0.4312**	**0.2419**	**0.1353**	0.0713	**0.1761**	**0.1169**
secondary-base	25	0.4191	0.2331	0.1327	0.0752	0.1384	0.1196
secondary-ft	1	0.3457	0.1934	0.1092	0.0610	0.1507	0.1096

in a warm-up phase and decrease it afterwards proportionally to the inverse square root of the current training iteration i

$$\eta = d_{\text{model}}^{-0.5} \cdot \min(i^{-0.5}, i \cdot w^{-1.5}). \quad (1)$$

In contrast to the original Transformer architecture, we used $w = 10,000$ for the number of warm-up steps.

For the base model of our primary model (*primary-base*), we observed the best validation performance on TRECVID-VTT after 25 epochs with a CIDEr score of 0.18. We used this model to finetune on only on the TRECVID-VTT dataset (*primary-ft*). In doing so, we slightly improved the scores as can be seen in Table III. For our second model, we chose the same approach but trained the base model on more data sources, namely MSR-VTT, AC-GIF and 90% of TRECVID-VTT. The best scores were also observed after 25 epochs and are in the same range as our primary model. However, when finetuning the second model on MSR-VTT only, the scores constantly decreased expect the CIDEr score as can bee seen in Table III. We submitted results generated by the second finetuned models, because we selected it based on the CIDEr scores and the generated captions on the validation set looked quite promising.

VI. RESULTS

TABLE IV
SUBMITTED MODELS AND THEIR RESPECTIVE PERFORMANCE ON THE UNSEEN TEST DATASET.

Model	BLEU	CIDEr	CIDEr-D	METEOR
primary-ft	0.018	0.140	0.064	0.202
second-ft	0.011	0.136	0.060	0.204

Before switching to the Transformer model, we experimented on a vanilla captioning model with an encoder and a LSTM decoder similar to [5]. These experiments have shown to deliver very bad performance and very similar sentence. Hence, we switched to the Transformer architecture, which yielded better results both qualitatively and quantitatively. For the TRECVID workshop [10], we submitted captions generated on the provided test videos $(1,700)$ for both of our models.

These captions were evaluated by the workshops organizers. Compared to our validation set scores, the evaluation on the test set yields worse results as can be been in Table IV. Especially, the BLEU score is much lower on the test data than on the evaluation data. However, the METEOR score is better on the test set.

We depict four videos and their generated caption in Figure 3. We see that for the first three videos our generated captions match the video content quite good. The first video does indeed look like a man talking to people in a classroom. Only if we look closer, we see that this is not a classroom, but rather some presentation in front of adults. In the second video, our model detects a young man singing and fails to recognize that there are two men, one of which is singing and the other is playing the piano. In the third video, our model detects a group of people who seem to be dancing. But it places them on a beach rather than in a pedestrian zone. In the fourth video, our model wrongly assumes that there is snow in a parking lot. However, it recognizes a man moving on the street in the daytime, but not the bicycle.

VII. CONCLUSION

In this notebook paper, we presented our VTT model based on a Tansformer [4] architecture. By extracting features for every frame of the videos, we were able to adapt the Transformer architecture to use videos in the encoder block. The decoder block could be used without modification. In addition, we modified the Multi-Head Attention of the encoder to use memory vectors similar to [7] which allow to memorize a priori knowledge about relationships between video frames. By training on multiple datasets, we are able to generate captions that describe video contents (see Figure 3). However, as not all objects and circumstances of the videos are detected and described correctly, we want to address object and relationship detection in future work.

REFERENCES

[1] Y. Pan, Y. Li, J. Luo, J. Xu, T. Yao, and T. Mei, "Auto-captions on gif: A large-scale video-sentence dataset for vision-language pre-training," *arXiv preprint arXiv:2007.02375*, 2020.

[2] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

[3] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, *et al.*, "Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval," 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016.

[7] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10578–10587, 2020.

[8] P. Das, C. Xu, R. F. Doell, and J. J. Corso, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2634–2641, 2013.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

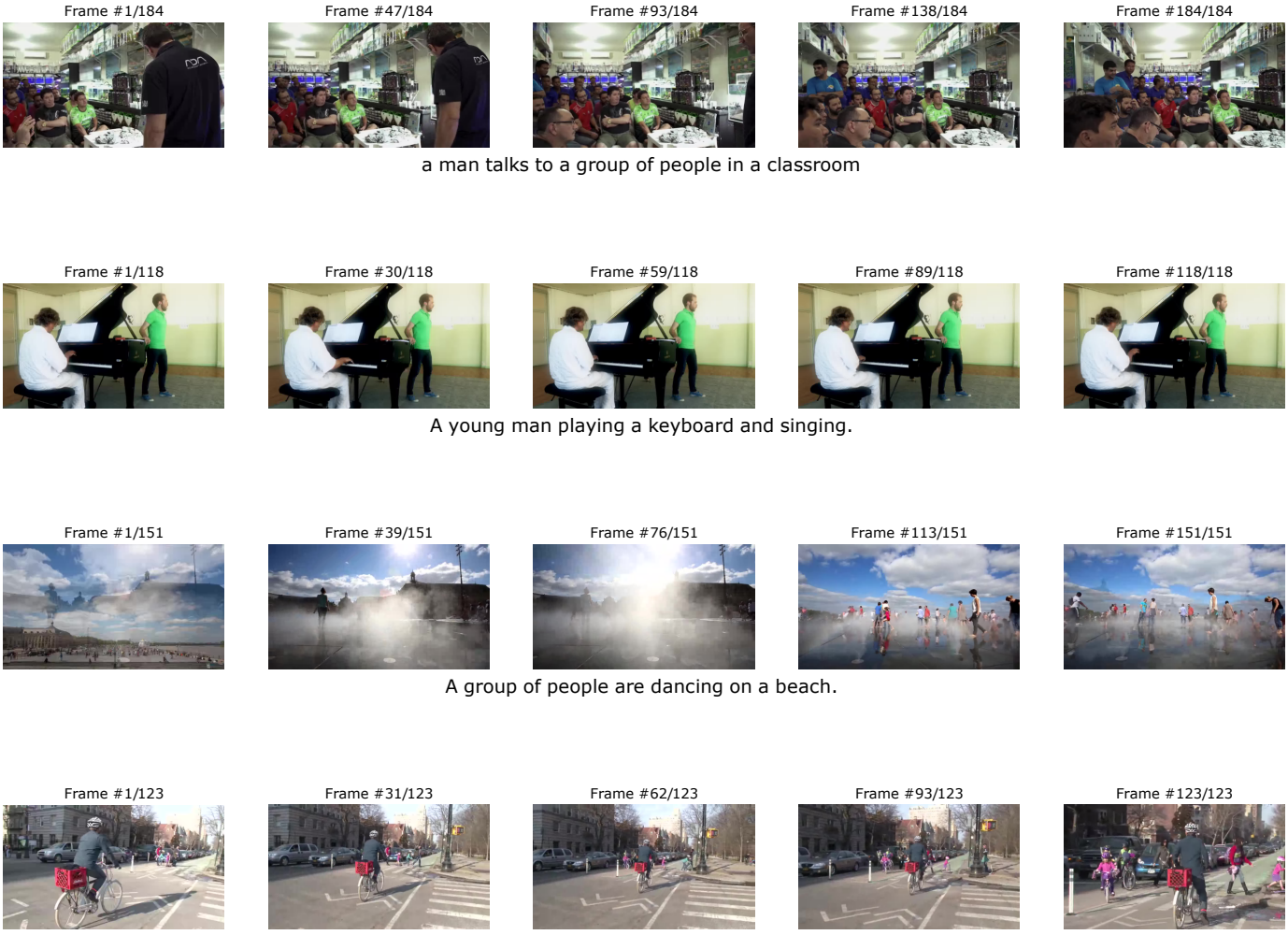

Fig. 3. Four videos from the test dataset and the corresponding captions generated by our model **primary-ft**.

[10] G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, J. Zhang, E. Godard, L. Diduch, J. Liu, A. F. Smeaton, Y. Graham, G. J. F. Jones, W. Kraaij, and G. Quénot, "Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains," in *Proceedings of TRECVID 2020*, NIST, USA, 2020.