# Synchronized audio-visual frames with fractional positional encoding for transformers in video-to-text translation

**Philipp Harzig, Moritz Einfalt, Rainer Lienhart**

**Angaben zur Veröffentlichung / Publication details:**

Harzig, Philipp, Moritz Einfalt, and Rainer Lienhart. 2022. "Synchronized audio-visual frames with fractional positional encoding for transformers in video-to-text translation." In *2022 IEEE International Conference on Image Processing (ICIP), 16-19 October 2022, Bordeaux, France*, edited by Yannick Berthoumieu, Pascal Frossard, Giuseppe Valenzise, and Thomas Maugey, 2041–45. Piscataway, NJ: IEEE. https://doi.org/10.1109/ICIP46576.2022.9897804.



# SYNCHRONIZED AUDIO-VISUAL FRAMES WITH FRACTIONAL POSITIONAL ENCODING FOR TRANSFORMERS IN VIDEO-TO-TEXT TRANSLATION

*Philipp Harzig* *Moritz Einfalt* *Rainer Lienhart*

University of Ausgsburg, Germany, {firstname.lastname}@uni-a.de

## ABSTRACT

Video-to-text (VTT) is the task of automatically generating descriptions for short audio-visual video clips. It can help visually impaired people to understand scenes shown in a YouTube video, for example. Transformer architectures have shown great performance in both machine translation and image captioning. In this work, we transfer promising approaches from image captioning and video processing to VTT and develop a straightforward Transformer architecture. Then, we expand this Transformer by a novel way of synchronizing audio and video features in Transformers which we call Fractional Positional Encoding (FPE). We run multiple experiments on the VATEX dataset and improve the CIDEr and BLEU-4 scores by 21.72 and 8.38 points compared to a vanilla Transformer network and achieve state-of-the art results on the MSR-VTT and MSVD datasets. Also, our novel FPE helps increase the CIDEr score by relative $8.6\,\%$.

***Index Terms*—** Video-to-text, Transformer, Positional Encoding, Synchronization, Audio-visual

## 1. INTRODUCTION

Since the introduction of the Transformer architecture by Vaswani et al. [23], massive improvements in the task of sequence transduction and machine translation have been made. Thus, it is natural to adapt this technique to image captioning [4, 12, 8, 28] and video-to-text (VTT). In this work, we address the video-to-text (VTT) task [30, 15, 6, 29, 7, 18, 25, 31, 13, 22] and start with a modified Transformer that is able to cope with video inputs. Then, we investigate several improvements by adopting various techniques from the domain of image captioning. Ultimately, we present a way to easily align video and audio features independent of their respective sampling rates. We align the features by extending the positional encoding to support fractional positions.

Our contributions are as follows: First, we develop a simple Transformer model for generating descriptions for short video clips. We reuse and adopt the best approaches from image captioning and present a modified learning rate schedule for our VTT Transformer. Second, we introduce Fractional Positional Encoding (FPE), an extension to the traditional positional encoding, which allows to synchronize video and audio frames independent on their respective sampling rate. By using FPE, we improve our CIDEr score by relative $8.6\,\%$. Furthermore, we achieve state-of-the-art scores on the MSVD and MSR-VTT datasets.

## 2. MODEL AND PROPOSED METHOD

We utilize a slightly modified Transformer [23] as our baseline model. The Transformer architecture is built around the idea of transforming sequences from one domain to another, i.e., the original Transformer is a machine translation model that operates on sequences of tokens (words). However, we work on a different input domain (i.e., video clips) instead of sentences. Thus, we modified the encoder of the original Transformer architecture by altering its inputs. We feed the encoder with video clip features. We embed these features and add the positional encoding on top of these embeddings in order to maintain information about absolute and relative ordering of the sequence. We use a learned word embedding to convert the input tokens to vectors of dimension $d_{\text{model}}$ and share the weight matrix with a learned linear projection layer to predict the probabilities of the next word at the end of the decoder [23, 19]. Given the embedded tokens and the encoder outputs, the decoder generates its output one word at a time. Similar to most encoder-decoder sequence models, the decoder uses the output of the previous step as input to the current step in an auto-regressive way when generating text. Thus, we simply optimize the cross-entropy loss for each word in every target sentence during training.

In our VTT model, we employ techniques and methods from the related task of image captioning and adapt the architecture to use video clips as inputs. Additionally, we introduce the novel Fractional Positional Encoding that allows to synchronize audio-visual frames in a Transformer encoder.

### 2.1. Baseline Model Configuration

**Inflated 3D ConvNet and VGGish.** For our input vision features, we use the well-known Inflated 3D ConvNet (I3D) [2] architecture to extract features from the input video clips. In particular, we extract features from the videos with the RGB-I3D model, which was pretrained on the Kinetics Human Action Video dataset [10]. We extract audio features with the

VGGish [9] architecture. We forward both the visual and audio features through a dense embedding layer to match the model's dimension $d_{\text{model}} = 512$.

**Memory-Augmented Encoder.** We make use of the memory-augmented encoding [4], which encodes multi-level visual relationships with a-priori knowledge. In our case, memory-augmented encoding allows to encode persistent a-priori knowledge about relationships between frames within each training video, which later can be transferred to unseen video samples.

**Subword and BERT Vocabulary.** Instead of implementing an ordinary dictionary that takes the $n_{\text{Voc}}$ most frequent words into account, we employ the WordPiece tokenizer [26]. The goal is to represent rare words by splitting them up into word-pieces, which can later be recovered. For our vocabulary, we use the default BERT [5] tokens.

**Learning-Rate Scheduling** We extend the default learning rate schedule from [23] with the SGDR (Stochastic Gradient Descent with Warm Restarts) [14] learning rate schedule. Initially, we found this technique to harm our final scores, i.e., the Transformer network did not seem to initialize correctly. However, when combining this approach with a warm-up phase, we did notice some improvements over the default Transformer learning rate schedule [23] (*schedule-default*). We depict *schedule-sgdr* alongside *schedule-default* in Figure 2.

### 2.2. Fractional Positional Encoding

We present a novel way of aligning vision and audio features within a Transformer model. The inputs to the Transformer consist of sequences of tokens extracted by the I3D and VGGish networks. As both the inputs to these feature extractors and the network architectures are different, these tokens are not synchronized: for vision features we extract I3D features without resampling the video and audio is resampled to 16 kHz. Thus, an I3D frame at a given position represents a different timestamp for videos with different framerates. If we resampled all videos to the same framerate, we would still have no way of synchronizing the vision frames with the audio frames, as those sampling rates differ. In other words, the audio frame at a given position would not match the timestamp of the I3D frame at the same position.

In the original work [23], the positional encoding has no inherent meaning other than to define the relative position of a word. For our input data however, vision and audio feature frames are aligned on the same time-axis and depend on their respective frame rate. Thus, we fix this problem by introducing the Fractional Positional Encoding (FPE) (see Figure 1). FPE is an extension to the traditional positional encoding that allows positional encoding on a fractional level. In order to fully utilize the audio features, the Transformer needs to know which audio frame corresponds to which vision frame. To do so, we calculate two timestamp factors for every video within

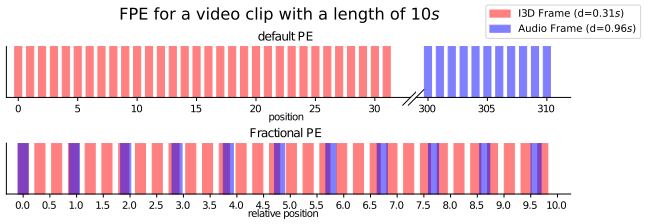


**Fig. 1**. The default positional encoding for audio and video frames (on top) in comparison with the FPE (bottom) for an exemplary video. The video has 32 I3D frames and 11 audio frames. The lengths ($d$) of audio and video frames differ.

the dataset, i.e., an audio and a vision timestamp factor. Both timestamp factors indicate the number of seconds each frame lasts. We then multiply the integer indices of each frame with the corresponding timestamp factor. Thus, we ensure that audio and video frames are properly aligned relative to their timestamp.

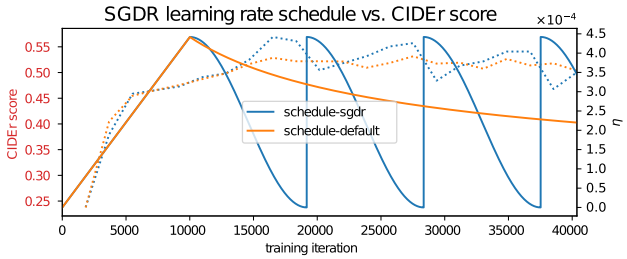


**Fig. 2**. Course of learning rate plotted against the CIDEr validation score of models *audio* and *audio-sgdr*. We plotted the learning rates with solid lines and the corresponding validation scores with a dotted line style.

### 2.3. Self-Critical Sequence Training

In our baseline model, we optimize the objective of maximizing the likelihood of the next ground-truth word given previous ground-truth words and the encoder outputs. This approach is called "Teacher-Forcing" [1] and has a serious drawback, i.e., the training phase is different from the inference phase (*exposure bias* [20]). In addition, our models are trained with cross-entropy loss and evaluated with non-differentiable metrics (e.g., CIDEr [24] and BLEU [17]). Therefore, we utilize Self-Critical Sequence Training [21] (SCST), which is a variation of the popular REINFORCE algorithm that utilizes the outputs of the model's test-time inference algorithm: First, we greedily sample a baseline caption for each video clip with our model in inference mode. Second, we sample 5 sentences for the corresponding video clip in training mode using monte-carlo sampling. Then, we calculate the CIDEr and BLEU-4 scores for the baseline

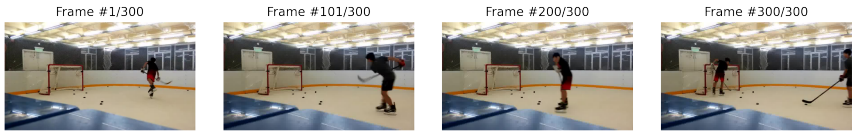


**Fig. 3**. Examples of generated descriptions for an example video from the validation split. We see four frames from each video together with the frame number on the left and the generated caption for each model on the right.

caption and subtract it from scores of the sampled captions. Thus, sampled captions with a higher CIDEr score or BLEU-4 score than the baseline caption get a positive reward and vice versa. By optimizing for this objective, sampled captions with a higher CIDEr score will be increased in probability, while we try to make bad captions less likely.

## 3. EXPERIMENTS

### 3.1. Datasets, Preprocessing and Implementation Details

**Datasets.** We mainly use the VATEX Dataset [25] for our experiments. The VATEX dataset is split into 4 sets: the training set, the validation set, the public test set and the private test set. Additionally, we train our final models on the MSR-VTT [27] and MSVD [3] datasets. For MSVD, we follow the common practice and split the 1970 available video clips into three partitions of 1200, 100 and 670 for training, validation and test, respectively. For MSR-VTT, we use splits containing 90 %, 5 % and 5 %.

**I3D Features.** We extract video clip features with the RGB-I3D pretrained on the Kinetics Human Action Video dataset [10]. The I3D yields features of dimension $\mathbb{R}^{n_v \times 1024}$, where in this case $n_v$ is the number of I3D frames. Furthermore, we learn an embedding layer to embed the I3D features into the model dimension ($\mathbb{R}^{n_v \times 512}$).

**Audio features.** We take the audio of the video, resample it to 16 kHz and extract features with the VGGish [9] network. This network yields features of dimension $\mathbb{R}^{n_a \times 128}$. Here, $n_a$ is the number of audio features, which is different from $n_v$.

**Implementation details.** Our model is implmented with TensorFlow 2 and we publish our code on GitHub[1]. As a baseline model, we implement a vanilla Transformer [23] model with $d_{\text{model}} = 512, d_{\text{ff}} = 2048$. Our encoder and decoder each have $N = 8$ layers with $h = 8$ parallel attention heads. We also adopt the same learning rate schedule from [23], however, we change the number of warm-up steps to 10,000. As optimizer, we use Adam [11] with the learning rate schedules from Section 2.1. We train for a maximum number of 50 epochs with a batch size of 128 and employ early stopping based on the validation CIDEr score. For fine-tuning with self-critical learning, we lower the effective batch size to 16 (i.e. 4 GPUs with batch size 4) and use a constant learning rate of $\eta = 5 \cdot 10^{-6}$.

[1] https://github.com/philm5/fpe-vtt

### 3.2. Discussion of Results

In the following, we discuss the results of the extensions presented in Section 2. In Table 1, we depict results on the validation set of the VATEX dataset. In Figure 3, we show generated captions for every model for an example video from the VATEX validation set. Both when looking at the scores and the generated descriptions, we see that our *baseline* model scores worst across all metrics. The baseline model only uses features from the RGB-I3D network with an image embedding layer for the encoder.

**Memory-Augmented Encoder.** Adding a memory vector to the key and value of the multi-head self-attention allows the encoder network to learn a-priori knowledge about relationships on an intra-frame level. For example, when we look at sentences generated for the video clip in Figure 3, we see an ice hockey player doing some shots on a goal. Comparing the caption generated by the *memvec* model to the captions of the baseline model, we see the model has memorized that ice hockey often is played within an *ice rink*. In addition, we see a slight boost in all scores.

**Learning Rate Scheduling.** Replacing the default Transformer learning rate schedule with our modified version of SGDR (*audio-sgdr*) improves the performance by 3.76 points and 1.81 points in CIDEr and BLEU-4 in contrast to *audio*, respectively. As we have already discussed in Section 2.1, the fast decay of the SGDR schedule helps to boost our validation scores as we depict in Figure 2. After the warm-up phase of 10,000 steps, the validation accuracy makes another climb until it hits its maximum CIDEr score of 56.92 at the end of the first decay.

**FPE.** In contrast to a naïve concatenation of audio and video features (*audio-sgdr*), FPE (*audio-sgdr-FPE*) boosts performance across all metrics significantly. Most notably, synchronizing audio and video features by their relative position has the largest benefit on the CIDEr metric, where we gain 4.88 points. Even during self-critical fine tuning (see next paragraph), FPE (*SCST-Cider-B4-FPE*) achieves improvements across all metrics. Thus, we conclude that FPE is an easy and effective way to synchronize audio and video features in Transformers.

**SCST.** We initialize the self-critical sequence training with the best models *audio-sgdr* and *audio-sgdr-FPE*. As reward function, we calculate the CIDEr and BLEU-4 scores of the baseline caption and the sampled sentences. We see that di-

| Model | Features | \|mv\| | FPE | lr Schedule | B@1 | B@2 | B@3 | B@4 | M | R | C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **baseline** | I3D | 0 | — | Default | 69.64 | 52.49 | 38.94 | 27.92 | 20.85 | 46.33 | 49.13 |
| **memvec** | I3D | 64 | — | Default | 71.12 | 53.83 | 39.79 | 28.26 | 21.76 | 47.14 | 51.38 |
| **audio** | I3D+VGGish | 64 | — | Default | 71.39 | 55.03 | 41.56 | 30.35 | 22.06 | 47.90 | 53.16 |
| **audio-sgdr** | I3D+VGGish | 64 | — | sgdr | 73.53 | 57.55 | 43.81 | 32.16 | 22.70 | 49.07 | 56.92 |
| **audio-sgdr-FPE** | I3D+VGGish | 64 | ✓ | sgdr | 75.35 | 58.58 | 44.30 | 32.43 | 23.81 | 49.60 | 61.80 |
| **SCST-Cider-B4** | I3D+VGGish | 64 | — | $5 \cdot 10^{-6}$ | 78.21 | 61.03 | 46.26 | 33.92 | 23.65 | 49.91 | 68.62 |
| **SCST-Cider-B4-FPE** | I3D+VGGish | 64 | ✓ | $5 \cdot 10^{-6}$ | 78.74 | 62.82 | 48.64 | 36.30 | 24.52 | 51.91 | 70.85 |

**Table 1**. Ablation study for our VTT Transformer models on the VATEX validation set. On the left, we list the model names with their respective configurations (|mv|=size of the appended memory vector). On the right we list the validation scores (B@x=BLEU-x, M = METEOR, R = Rouge-L, C = CIDEr).

| | | Features | | | | MSVD | | | | MSR–VTT | | | | VATEX | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Year** | **I** | **M** | **O** | **A** | **B@4** | **M** | **R** | **C** | **B@4** | **M** | **R** | **C** | **B@4** | **M** | **R** | **C** |
| **ORG-TRL** | CVPR 2020 [30] | ✓ | ✓ | ✓ | — | 54.3 | 36.4 | 73.9 | 95.2 | 43.6 | 28.8 | 62.1 | 50.9 | 32.1 | 22.2 | 48.9 | 49.7 |
| **LSTM-TSA$_{IV}$** | CVPR 2017 [15] | | | | | 52.8 | 33.5 | — | — | — | — | — | — | — | — | — | — |
| **aLSTMs** | IEEE ToM 2017 [6] | ✓ | ✓ | — | — | 50.8 | 33.3 | — | — | 38 | 26.1 | | 43.2 | — | — | — | — |
| **RCG** | CVPR 2021 [29] | ✓ | ✓ | — | — | — | — | — | — | 42.8 | 29.3 | 61.7 | 52.9 | 33.9 | 23.7 | 50.2 | 57.5 |
| **NSA** | CVPR 2020 [7] | — | ✓ | ✓ | — | — | — | — | — | — | — | — | — | 31.4 | 22.7 | 49 | 57.1 |
| **SemSynAN** | CVPR 2021 [18] | ✓ | ✓ | — | — | **64.4** | **41.9** | **79.5** | **111.5** | **46.4** | **30.4** | **64.7** | 51.9 | — | — | — | — |
| **VATEX** | CVPR 2019 [25] | — | ✓ | — | — | — | — | — | — | — | — | — | — | 28.7 | 21.9 | 47.2 | 45.6 |
| **SCST-Cider-B4-FPE** | Ours | — | ✓ | — | ✓ | 51.22 | 34.73 | 72.69 | 103.2 | 45.91 | 30.25 | 64.12 | **62.11** | 33.28 | 22.74 | 49.56 | 54.63 |
| Non peer-reviewed papers: | | | | | | | | | | | | | | | | | |
| **MV+HR** | arXiv 2019 [31] | ✓ | ✓ | ✓ | — | — | — | — | — | — | — | — | — | **40.7** | **25.8** | **53.7** | **81.4** |
| **MM-Feat** | arXiv 2020 [13] | ✓ | ✓ | ✓ | ✓ | — | — | — | — | — | — | — | — | 39.2 | 26.5 | 52.7 | 76 |
| **NITS-VC** | arXiv 2020 [22] | — | ✓ | — | — | — | — | — | — | — | — | — | — | 22 | 18 | 43 | 27 |

**Table 2**. Comparison on VATEX, MSVD and MSR-VTT datasets against state-of-the-art methods. For VATEX, we tested our model on the private test set with the evaluation server. For MSVD and MSR-VTT, we use the test-splits discussed in Section 3.1. I, M, O and A denote image, motion, object and audio features.

rectly optimizing the common sentence metrics leads to big gains in the CIDEr metric, i.e., 68.62 points vs. 56.92 points. When combining SCST with FPE, our model produces the best results across all experiments and we improve by another 2.23 and 2.38 in CIDEr and BLEU-4, respectively.

### 3.3. Comparison with State-of-the-Art

We were not able to download all video files for the VATEX dataset from YouTube, thus we could not train, validate and test on the whole dataset. Additionally, we do not have audio features for those missing videos. Submitting generated descriptions to the evaluation server requires descriptions for every single of the 6,278 videos, thus, we use the VATEX authors' I3D features with no audio features for submitting results. In Table 2, we depict results of our model *SCST-Cider-B4-FPE* trained in the same manner on both train and validation splits. Our model scores not as well as the models from the VATEX video captioning challenge such as the models from Zhu et al. [31] and Lin et al. [13], who use ensembles of up to 32 models. However, across all published works on video captioning, we achieve similar performance on the reported metrics. We also train our model on the MSVD and MSR-VTT datasets to prove the effectiveness of our method. On the MSVD dataset, our scores are below SemSynAN [18] but otherwise better than all other methods listed in Table 2. For MSR-VTT, however, our final model outperforms SemSynAN by 10.21 points in CIDEr and performing similar to it for the other metrics.

## 4. FUTURE WORK AND CONCLUSION

In our work, we presented a Transformer-based video-to-text architecture aimed to generate descriptions for short videos. Utilizing the best approaches from the related field of image captioning, we designed an architecture that generates appropriate and matching captions for video clips. Furthermore, we introduced the novel Fractional Positional Encoding to properly synchronize video and audio features with different sampling rates, which significantly improves results across all metrics. In combination with self-critical sequence training, we were able to considerably boost the performance of a baseline model by an absolute of 21.72 points or relative 144% in the CIDEr metric.

In the future, we want to expand our model with the X-Linear Attention block [16], which shows huge potential in other works [31]. Furthermore, we will extend the model by a multi-modal training objective that takes Chinese captions from the VATEX dataset into account in order to improve training feedback.

## 5. REFERENCES


[1] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, et al. Scheduled sampling for sequence prediction with recurrent neural networks. *arXiv preprint arXiv:1506.03099*, 2015. 2

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE CVPR*, pages 6299–6308, 2017. 1

[3] David Chen and William B Dolan. 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, 2011. 3

[4] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, et al. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE CVPR*, pages 10578–10587, 2020. 1, 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 2

[6] Lianli Gao, Zhao Guo, Hanwang Zhang, et al. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017. 1, 4

[7] Longteng Guo, Jing Liu, Xinxin Zhu, et al. Normalized and geometry-aware self-attention network for image captioning. In *Proceedings of the IEEE CVPR*, pages 10327–10336, 2020. 1, 4

[8] Sen He, Wentong Liao, Hamed R Tavakoli, et al. Image captioning through image transformer. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1

[9] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 2, 3

[10] Will Kay, Joao Carreira, Karen Simonyan, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 3

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[12] Guang Li, Linchao Zhu, Ping Liu, et al. Entangled transformer for image captioning. In *Proceedings of the IEEE ICCV*, pages 8928–8937, 2019. 1

[13] Ke Lin, Zhuoxin Gan, and Liwei Wang. Multi-modal feature fusion with feature attention for vatex captioning challenge 2020. *arXiv preprint arXiv:2006.03315*, 2020. 1, 4

[14] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[15] Yingwei Pan, Ting Yao, Houqiang Li, et al. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE CVPR*, pages 6504–6512, 2017. 1, 4

[16] Yingwei Pan, Ting Yao, Yehao Li, et al. X-linear attention networks for image captioning. In *Proceedings of the IEEE CVPR*, pages 10971–10980, 2020. 4

[17] Kishore Papineni, Salim Roukos, Todd Ward, et al. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 2

[18] Jesus Perez-Martin, Benjamin Bustos, and Jorge Pérez. Improving video captioning with temporal composition of a visual-syntactic embedding. In *Proceedings of the IEEE/CVF WACV*, pages 3039–3049, 2021. 1, 4

[19] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, 2017. 1

[20] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, et al. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 2

[21] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, et al. Self-critical sequence training for image captioning. In *Proceedings of the IEEE CVPR*, pages 7008–7024, 2017. 2

[22] Alok Singh, Thoudam Doren Singh, and Sivaji Bandyopadhyay. Nits-vc system for vatex video captioning challenge 2020. *arXiv preprint arXiv:2006.04058*, 2020. 1, 4

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2, 3

[24] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE CVPR*, pages 4566–4575, 2015. 2

[25] Xin Wang, Jiawei Wu, Junkun Chen, et al. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE ICCV*, pages 4581–4591, 2019. 1, 3, 4

[26] Yonghui Wu, Mike Schuster, Zhifeng Chen, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 2

[27] Jun Xu, Tao Mei, Ting Yao, et al. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE CVPR*, pages 5288–5296, 2016. 3

[28] Jun Yu, Jing Li, Zhou Yu, et al. Multimodal transformer with multi-view visual representation for image captioning. *IEEE transactions on circuits and systems for video technology*, 30(12):4467–4480, 2019. 1

[29] Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, et al. Open-book video captioning with retrieve-copy-generate network. In *Proceedings of the IEEE CVPR*, pages 9837–9846, 2021. 1, 4

[30] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, et al. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE CVPR*, pages 13278–13288, 2020. 1, 4

[31] Xinxin Zhu, Longteng Guo, Peng Yao, et al. Vatex video captioning challenge 2020: Multi-view features and hybrid reward strategies for video captioning. *arXiv preprint arXiv:1910.11102*, 2019. 1, 4