Zero-Shot Video Captioning by Evolving Pseudo-tokens

Yoad Tewel
yoadtewel@mail.tau.ac.il
Yoav Shalev
itsyoavshalev@gmail.com
Roy Nadler
royxnadler@gmail.com
Idan Schwartz
idanschwartz@gmail.com
Lior Wolf
wolf@cs.tau.ac.il

School of Computer Science,
Tel Aviv University,
Tel Aviv, Israel.

Abstract

We introduce a zero-shot video captioning method that employs two frozen networks: the GPT-2 language model to generate sentences and the CLIP to maintain a high average matching score between the generated text and the video frames. Existing zero-shot captioning methods use token-level optimization that drives the generation of each token to be related to the image. However, maintaining language fluency with a set of frames can be challenging since (i) a single token has to describe a set of non-homogeneous frames, and (ii) the generation may commit to a single direction, restricting the flexibility of the process. In our approach, we use pseudo-tokens that update after each complete sentence is generated, gradually improving the specificity and comprehensiveness of the sentence while letting the user control the level of specificity. The optimization takes into account the whole sentence and does not require beam-searching. Our experiments show that the generated captions are fluent and display a broad range of real-world knowledge for both videos and images. Moreover, while current supervised video captioning methods generate captions that often follow a short and generic pattern based on the datasets they were trained on, our approach generates diverse and descriptive captions that are much more appealing to humans. Our code is available at: https://github.com/YoadTew/zero-shot-video-to-text.

1 Introduction

Image captioning is becoming increasingly accurate. However, the progress in video captioning is slower due to both methodological reasons and dataset construction challenges. First, the video captioning task itself is more ambiguous than image captioning. For example, do we want a complete description of the events in the video or a general description of it? Second, even much more limited tasks, such as action recognition in pre-trimmed videos,

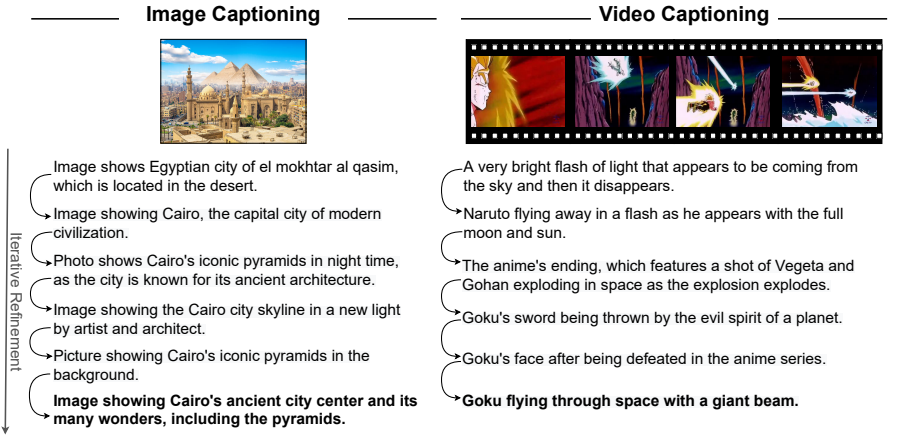


Figure 1: We present a novel way of optimizing a sentence generation process to match a set of images by using pseudo-tokens and iteratively generating sentences. Generation is done in a zero-shot manner and exhibits real-world knowledge utilizing CLIP and GPT-2 as knowledge sources. The narrative evolves through the iterations and tends to become increasingly specific. Optimization takes place after complete sentence generation. Notably, when generated without interference, the PLM generates low perplexity sentences.

remain technologically challenging. Third, the descriptions attached to web videos are often not an accurate depiction of the events of the video.

These challenges mean that strategies used in related tasks are less suitable for the task of video captioning. (i) One cannot obtain large datasets with reasonable noise levels. (ii) Learning on a carefully curated dataset would be too restrictive in terms of the obtained coverage. (iii) Relying on pre-trained action recognition models is not viable.

We, therefore, introduce a zero-shot method, which does not rely on video training data. It uses the information stored in two pre-trained and frozen networks to perform the video captioning task. One model is an autoregressive language model that can generate natural and mostly logical sentences. The second model is an image-text matching model that is used to steer the language model toward sentences that match a set of input frames.

Existing zero-shot captioning methods [42, 45] optimize each token individually during the autoregressive process, necessitating an early commitment to a narrative. While this approach performs well for images, we have found it unsuitable for video captioning. Our experiments demonstrate that sentence-level optimization is better suited to handle signals originating from the non-homogeneous frames of a video.

To address this, we generate a new caption at each iteration and optimize the pseudo-tokens using the signal obtained from the entire caption of that iteration. The generative process of our method involves a prompt consisting of three parts: (i) pseudo-tokens that are vectors in the latent space of the language models [16, 23], (ii) a random prompt such as "Image of" that provides context for the captioning task, but also varies ("Photo of", "Video of", etc.) as a form of inference-time augmentation, and (iii) the previously generated tokens.

Due to the autoregressive nature of the sentence generation process, the initial words of the generated sentence employ pseudo-tokens that were not optimized based on the signal obtained from the entire sentence. To address this limitation, we repeat the process and

initiate the autoregressive process with the pseudo-tokens obtained at the end of the previous generation iteration. This iterative approach leads to increasingly concrete prompts, as illustrated in Fig. 1.

In our implementation, we use the GPT-2 language model [33], due to its availability and the CLIP image-text matching model [34], which is often used in zero-shot learning. We experiment with both video captioning and describing image sets. Our results show a clear advantage over the state-of-the-art video captioning methods and over recent zero-shot image captioning methods. Furthermore, our approach has two main efficiency advantages: first, it does not require beam-search. Second, the optimization generates multiple sentences with increasing details, providing control over the level of specificity (the intermediate stages of token-level optimization are partial sentences). Our code is attached as a supplementary.

2 Related Work

Visual captioning is a fundamental vision and language task. Early methods applied RNNs [13, 25]. Attention was added to identify relevant salient objects [36, 53]. Graph neural networks and transformers helped model spatial and semantic interactions [12, 54, 55]. Other video-based tasks include action recognition [41], paragraph captioning [56], and video object segmentation [51]. This work considers video captioning by generating a single sentence that adequately describes a set of frames. Despite the lack of temporal information in a set, we find that the language model generates a logical order of events.

In various contributions, sparse sampling along with better spatial reasoning has proved sufficient for handling reasoning tasks, such as video dialogs [37] and video retrieval [14, 15, 35, 58]. An attention module that selects the relevant frames can reduce the temporal dimension [1, 8]. In our work, we also employ sparse sampling that utilizes distances in the CLIP embedding space to construct a set of the most relevant frames.

Significant improvements have been achieved by using large-scale unsupervised vision-language data sets with millions of image-text pairs [7, 21, 57] and videos [24, 27, 59]. The unsupervised data is used in a pre-training phase. Fine-tuning for a particular task is done in the final stage, using smaller datasets annotated by humans, leading to dull sentences that present the same repetitive patterns even for significantly different baseline methods [5, 32].

CLIP is trained on 400M images/sentence pairs from the web [34], resulting with a powerful text-image matching score by learning to project them to a shared embedding space. Matching videos to text also benefited from a contrastive approach [17, 50]. However, video data can be challenging to collect. In our case, we used CLIP to guide a language generator in a zero-shot manner.

While several zero-shot tasks, such as image classification and action recognition, benefited from CLIP's matching score, generative tasks based directly on the score are rare, since the score requires seeing both text and image. Instead, multiple contributions rely on CLIP's image and text encodings, which are known to improve performance in vision+language tasks [39], especially in image and video captioning [28, 44]. However, fine-tuning distorts the latent semantics of CLIP's encoder [45]. MAGIC employs CLIP scores to shift PLM logits towards image correspondence [42]. Despite this, they fine-tune the PLM on the text corpus of MS-COCO captions, so robustness is still compromised. Alternatively, CLIP can be used as part of a loss term to guide generative processes to match language and text, for example, as a loss term for 3D mesh generation [26] or text-guided image generation [4, 30].

Recently, it was suggested to use CLIP loss to guide a Pretrained Language Model (PLM)

for image captioning [45]. Their method optimizes each generated token individually, aiming to obtain the token closest to the given image. In contrast, our method optimizes pseudo-tokens through iteratively generating sentences, aiming to steer the generation process of the entire sentence. Although their process is effective in describing one visual cue, it is challenged by the more difficult task of describing multiple images coherently. We demonstrate that manipulating an entire sentence without committing to a single-generation path is essential to video captioning. Optimizing an entire sentence means not requiring sequence generation strategies like beam search.

The literature on tuning prior knowledge within large-scale PLMs, such as GPT-2 [33], is growing rapidly. In this work, we present a novel PLM decoding approach that combines steering[6] and prompt tuning by generating sentences iteratively and applying prompt tuning [9, 16, 20, 23, 40, 48].

3 Method

Our goal is to create a sentence $S = \{t_1, \ldots, t_M\}$ of length M that describes a set of video frames $\mathcal{F} = \{F_1, \ldots, F_N\}$, where N is the number of frames. When $N = 1$, the problem corresponds to traditional image captioning.

Two components are at the core of our solution. The first is a pre-trained language model (PLM) that generates sentences, for which we use GPT-2. The second, CLIP, is a pre-trained model that computes the distance between a frame F and a sentence S, and guides the PLM during inference.

Guiding a PLM with CLIP has recently shown promising results for image captioning [43, 45]. These approaches use CLIP to optimize the next token to fit the image. We call this technique *token-level optimization*. However, when optimizing each token separately, language fluency may be compromised in the case of videos because each token often has to describe multiple non-homogeneous frames. Moreover, the generation commits to a single direction, restricting the flexibility of the process. By contrast, rather than optimizing tokens, our method performs a *sentence-level optimization*. To achieve this, the inference starts with randomly initialized *pseudo-tokens*. These tokens do not need to be actual words in the dictionary, but rather hidden states of words that can be optimized using gradient descent. Description of visual content is driven using *prefix-tokens*, such as 'Video showing'. The next step consists of generating multiple sentences and continuously optimizing the pseudo-tokens. This is accomplished by calculating two types of losses: (i) $\mathcal{L}_{\text{vision}}$, which is the sum of the distance between all frames in $\mathcal{F}$ and the generated sentence, and (ii) $\mathcal{L}_{\text{language}}$ which takes into account language characteristics by considering the PLM token distribution. With no additional supervision or training, we benefit from the extensive knowledge embedded in CLIP and GPT-2. Our autoregressive process is depicted in Fig. 2.

PLM Guidance with Prompt Learning PLMs are trained on vast web knowledge to optimize a sum of conditionals, i.e., $\max_\theta p(S) = \sum_{i=1}^{L} p_\theta(w_i|w_{1:i-1})$, where θ are trainable weights. The likelihood of each sub-sentence depends on its context. Thus, one can solve various tasks by altering the input context. For instance, to answer the question "What is the capital of Britain?" one could plug into the PLM the prompt "The capital of Britain is." The PLM then finds the most likely next token ("London") to optimize the conditional probability.

Prompt engineering entails finding the most suitable prompt for a given task. In our case, the task is to generate a sentence that maximizes similarity to a set of frames $\mathcal{F}$. The

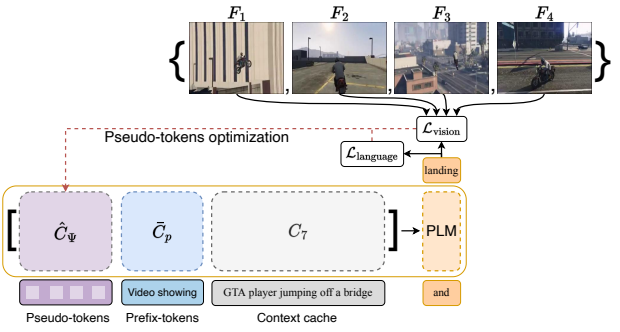


Figure 2: Illustration of our method for guiding a PLM to generate the word 'landing'. The pseudo-tokens ($\hat{C}_\Psi$) are optimized after a complete sentence is generated. Two signals steer the pseudo-tokens' representations, visual correspondence ($\mathcal{L}_{\text{vision}}$) and language fluency ($\mathcal{L}_{\text{language}}$).

similarity is measured in terms of CLIP's distance metric between image and text. Any image imposes its own set of constraints, and the prompt needs to account for all of them. The prompt must be flexible enough, which is ensured by optimizing pseudo-tokens, i.e., instead of finding real tokens for each video, we tune representative embeddings of tokens.

The GPT-2 PLM is built with L layers of Transformers, each composed of key and value embeddings, to model interactions between tokens [46]. The context of previous tokens can be cached by keeping their key and value representations. We denote the cache with $C_i = [K_j^l, V_j^l]_{j<i, l\leq L}$, where i is the number of tokens, and K_j^l, V_j^l are the key and value embeddings of the l-th Transformer layer of the j-th token.

Our method starts the autoregression process with a randomly initialized cached pseudo-prompt context, i.e., $\hat{C}_\Psi = [\hat{K}_j^l, \hat{V}_j^l]_{0<j<\Psi, l\leq L}$, which represents Ψ pseudo-tokens.

We further include $\bar{C}_p = [\bar{K}_j^l, \bar{V}_j^l]_{0<j<p, l\leq L}$, where p is the prefix length. The prefix-tokens serve to direct the task towards captioning a set of images. The prefix-tokens are sampled as one of the prompts in the set $\mathcal{P}$ ={"Image of", "Picture of", "Photo of", "Video of", "Image shows", "Picture shows", "Photo shows", "Video shows", "Image showing", "Picture showing", "Photo showing", "Video showing"}.

Overall, the autoregressive process takes the form

$$p_{i+1}(\hat{C}_\Psi) = \text{PLM}(t_i, [\hat{C}_\Psi, \bar{C}_p, C_i]), \tag{1}$$

where p_{i+1} is the distribution of the next token.

Loss: At each step in the auto-regression process, we aggregate our loss, which will be used for optimization only after generating a complete sentence. Our first loss term encourages the generated text to correspond to the set of images.

Let S_k be the sentence generated up until this stage, ending with the token k. The visual-semantic loss calculates the cross-entropy (CE) between the optimized PLM distribution and the CLIP potential similarity distribution θ_{CLIP}:

$$\mathcal{L}_{\text{vision}}(\hat{C}_\Psi) = \text{CE}\left(p_{i+1}(\hat{C}_\Psi), \theta_{\text{CLIP}}\right), \tag{2}$$

where $\theta_{\text{CLIP}}(k) \propto \sum_{F\in\mathcal{F}} \text{CLIP}(F, S_k)$ is the sum of CLIP's matching scores of S_k with all the frames in $\mathcal{F}$. We compute the score for the top 100 tokens according to the original PLM distribution, with the rest set to zero.

While the PLM is trained on natural text, the model in which the free-form context C_Ψ is added (Eq. 1) can shift to very different distributions during optimization. In order to maintain a fluent language, we define a language-related loss term,

$$\mathcal{L}_{\text{language}}(\hat{C}_\Psi) = \text{CE}\left(p_{i+1}(\hat{C}_\Psi), \text{PLM}(t_i, [\bar{C}_p, C_i])\right),$$

Dataset	Method	Supervised Metrics					Unsupervised Metrics			
		B@4	M	C	R	CLIP-S^{Ref}	CLIP-S	BLIP-S	Retrieval	PP
MSR-VTT	VNS-GRU [8]	0.453	0.299	**0.530**	0.634	0.739	0.626	0.623	0.446	118.81
	SemSynAN [34]	**0.464**	**0.304**	0.519	**0.647**	0.733	0.619	0.608	0.437	155.01
	Zero-Shot Methods									
	ZeroCap* [43]	0.023	0.129	0.058	0.304	0.739	0.710	0.575	0.442	54.71
	MAGIC* [42]	0.055	0.133	0.074	0.354	0.628	0.566	0.434	0.392	30.48
	Ours	0.030	0.146	0.113	0.277	**0.785**	**0.775**	**0.675**	**0.504**	**18.35**
MSVD	VNS-GRU [8]	**0.665**	**0.421**	**1.215**	**0.797**	0.780	0.673	0.646	0.557	418.72
	SemSynAN [34]	0.644	0.419	1.115	0.795	0.767	0.660	0.633	0.546	242.46
	Zero-Shot Methods									
	ZeroCap* [43]	0.029	0.163	0.096	0.354	0.762	0.765	0.642	0.500	28.44
	MAGIC* [42]	0.066	0.161	0.140	0.401	0.670	0.623	0.497	0.469	29.84
	Ours	0.030	0.178	0.174	0.314	**0.805**	**0.822**	**0.743**	**0.569**	**18.94**

Table 1: Quantitative results for video captioning. We separate the results into two categories: (i) supervised metrics that require human references, B@4 = BLEU-4, M = METEOR, C = CIDEr, R = ROUGE, and CLIP-S^{Ref}. (ii) Unsupervised metrics that use a pre-trained model, CLIP-S = CLIP-based image-text similarity, BLIP-S = BLIP-based image-text similarity, Retrieval = VideoCLIP-based video-text similarity, and PP = caption perplexity computed with BERT. (*) these are adapted from image captioning to video captioning.

which is the cross-entropy loss of the optimized PLM, as defined in Eq. 1, with the unmodified PLM distribution.

In order to have the generated text describe the set of images using fluent language, we solve the following optimization problem:

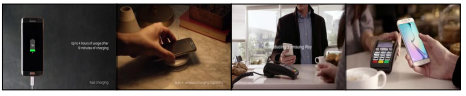
$$\min_{\hat{C}_\Psi} \mathcal{L}(\hat{C}_\Psi) = \min_{\hat{C}_\Psi} \mathcal{L}_{\text{vision}}(\hat{C}_\Psi) + \lambda \mathcal{L}_{\text{language}}(\hat{C}_\Psi)$$

where hyper-parameter λ calibrates the trade-off between relevance to the video and language fluency. The optimization process occurs during autoregression inference, generating sentences iteratively. We detail this process next.

Evolving Pseudo-Tokens Optimization The optimization occurs during the generation of the entire sentence, increasing image correspondence by applying iterations. Notably, we do not use annotations, nor are any parameters fine-tuned in a separate phase. We calculate the partial loss for each generated token and accumulate it. After a complete sentence is generated, indicated upon reaching a dot token, we perform one optimization step, i.e., $\hat{C}_\Psi \longleftarrow \hat{C}_\Psi + \alpha \frac{\nabla_{\hat{C}_\Psi} \mathcal{L}(\hat{C}_\Psi)}{\|\mathcal{L}(\hat{C}_\Psi)\|^2}$. With the optimized pseudo-tokens as the context, we begin a new sentence and continue the generation process. This optimization process is autoregressive in its nature, and, therefore, the first words are generated with preliminary context. Though early sentences usually have good language, each successive sentence increases the ability to ground objects. As a result, the process shifts from a general discussion to a more concrete explanation (see Sec. A.7 in the Appendix for qualitative examples).

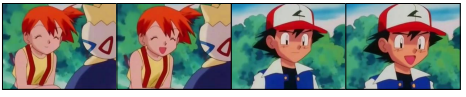
Throughout our experiments, we employ twenty iterations. Each time a new sentence is generated, we pick a prefix-token from the set $\mathcal{P}$ at random, which acts as data augmentation; see Appendix A.3 for an analysis of using a random prefix.

Adapting our model for video captioning begins by sampling three frames every second. To avoid repetition and capture diverse frames, we further subsample the frames with a CLIP-based strategy. See Appendix Fig. A.2 for more details and qualitative results.



VNS-GRU: A man is talking about a phone
SemSynAN: A man is talking about a phone
ZeroCap: A mobile phone payment in-wallet is shown.
MAGIC: A large commercial double decker bus on a city and the person is texting.
Ours: Samsung debit card charging in a smartphone advertisement, showing how the technology works.

VNS-GRU: A car is being displayed
SemSynAN: A car is being displayed
ZeroCap: A Toyota MotorAV receiver module display vehicle information in Audi software is uploaded to Google",
MAGIC: A black car is shown on a city street.
Ours: Audi's new navigation system in a video posted by carmaker.



VNS-GRU: A cartoon character is talking to another cartoon
SemSynAN: A cartoon character is talking to a man
ZeroCap: A conversation between between the character Pikachu and a a trainer is shown in the video.
MAGIC: A red and white commercial with a person on a baseball field.
Ours: Pokemon's character Ash talking to a girl in the background.

VNS-GRU: A man is talking
SemSynAN: A man is talking
ZeroCap: A 1983 Colbert audience is being restored.
MAGIC: A man smiling while holding a yellow frisbee.
Ours: The audience laughing and cheering at a joke about comedian's appearance in an interview with presenter.

Figure 3: Examples of our video captions with two types of baselines: (i) the supervised methods SemSynAN and VNS-GRU; and (ii) the zero-shot methods ZeroCap and MAGIC. Notably, our method grounds objects from different frames and exhibits real-world knowledge. The 1st and 2nd rows provide examples of real-world knowledge.

4 Results

To evaluate the quality of our captions, we run four types of experiments: (i) video captioning, (ii) image captioning, (iii) a stress test of unrelated image set captioning, and (iv) a text inversion method for CLIP's embedding space. We report the experiment settings, parameter sensitivity, and ablations in Appendix A.6.

We use two types of metrics: (i) Supervised metrics that measure text correspondence to human references: BLEU [29], METEOR [6], CIDEr [47], SPICE [2]. Lastly, CLIP-S$^{\text{Ref}}$ [11] measures semantic similarity by utilizing CLIP's textual encoder. (ii) Unsupervised metrics that are computed without referring to the human annotation. Relatedness to the visual cue is measured by averaging CLIP or BLIP [19] image similarity scores to the generated sentence across the frames. Relatedness to the video is measured by the VideoCLIP [51] video-to-text distance metric ("Retrieval" in the results table). Language quality is estimated using the perplexity score of the generated caption, employing BERT [7].

Two video datasets are used: MSR-VTT [52] and MSVD [49]. MSR-VTT is a large-scale dataset with about 50 hours of videos divided into 10,000 videos with 20 descriptions each. It includes a variety of categories, such as video games and TV shows. The test set consists of 2,990 videos. MSVD contains 1,970 short video clips, 670 of which are dedicated for testing. All experiments are carried out on the test set.

Quantitative Analysis: In Tab. 1, we compare our approach with supervised state-of-the-art baselines for video captioning. We also compare it with zero-shot video captioning baselines we created by modifying CLIP-based zero-shot image captioning methods: ZeroCap [45], a zero-shot method for image captioning, which also optimizes the generated sentence during inference. We adapt their method from image captioning to video captioning by replacing their single image CLIP loss with ours (i.e., a sum of CLIP losses for each frame). MAGIC [42], another zero-shot method for image captioning, which uses MAGIC scores, i.e., a CLIP-based measure of how closely a sentence ending with a given token matches an image, to skew the next-token distribution of a pre-trained language model to match a given image. To adapt their model to videos, we aggregate the CLIP score of all sampled frames to calculate the MAGIC score before applying a softmax.

As expected, the supervised models VNS-GRU [5] and SemSynAN [32] perform signifi-

cantly better on supervised metrics based on correspondence to human references. However, when considering semantic relatedness to annotations (i.e., CLIPScoreRef), our method is better (0.785 vs. 0.739 and 0.733). We next look at unsupervised metrics. BLIP-Score suggests that our text is more relevant to the frames (0.675 vs. 0.623 and 0.608). Furthermore, when considering the entire video temporally, with VideoCLIP text-to-video, our method has the best performance (0.504 vs. 0.446 and 0.437).

To understand the source of the weakness of the supervised methods, we measure the novelty of the generated sentences. Aggregated over the entire MSR-VTT test set, our method has a vocabulary size of 4,372. In contrast, SemSynAN and VNS-GRU use only 359 and 435 words, respectively, with roughly 40% of the generated sentences existing in the training set. Thus, since they are limited to vocabulary and styling from the training set, supervised methods do not demonstrate real-world knowledge and have high perplexity scores. In addition, as we show in the qualitative experiments, all supervised methods generate similar sentences despite having different architectures.

As for zero-shot methods, both ZeroCap and MAGIC fall short in all the unsupervised metrics: (i) the language fluency is compromised with token-optimization techniques (PP of 18.35 vs. 30.58). Note that these methods use beam-search decoding, while our method does not. We note that MAGIC employs a fine-tuned language model based on the text corpus of MS-COCO captions. Thus, while MAGIC is comparable to our method concerning the supervised metrics, it falls short on the unsupervised metrics that do not depend on curated human references. More evidence for this point is given in the qualitative analysis.

Qualitative Analysis: In Fig. 3, we show examples of our video captions. There are three main conclusions we draw: (i) Supervised methods are overfitted to the training data. Although they have different architectures, their generated captions are very similar. Moreover, their grounding capabilities are limited to relatively abstract objects. For example, they recognize a car or a phone in the first row. However, they miss the brand or the commercial intent of the video. (ii) The zero-shot methods, ZeroCap and MAGIC, fail in aggregating information from multiple frames. E.g., they do not describe the comedian or the interviewer in the right video on the 2nd row. The reason could lie in the challenge of optimizing a single token to relate to multiple frames. (iii) Our method grounds more specific details. MAGIC's captions are often broad, failing in videos that require real-world knowledge. For instance, our method detects brands (e.g., Samsung), and MAGIC only mentions 'texting.' Moreover, ZeroCap often identifies a related but wrong entity (e.g., Pikachu instead of Ash in the Pokemon video). More examples are available in Appendix A.7.

Image Captioning Tab. 2 compares our method to state-of-the-art image captioning approaches. Our method has a significantly better perplexity score (19.04 vs. 25.74). Our captions also show high relatedness to the image based on CLIP-based scores (0.798 vs. 0.778). The full study on image captions is available in Appendix A.3.

User study In Tab. 3, we evaluate the quality of zero-shot captioning methods on images and videos that require real-world knowledge. We asked 20 annotators to rank each caption based on three properties: human-like, grounding, and overall score. The test set included 10 images from the web and 10 videos from the MSR-VTT test set. Our approach performed significantly better for videos, with a mean opinion score of 4.14 compared to 2.30 for the baseline. We find ZeroCap struggles with generating human-like language (2.27 on human-like). Magic improves sentence quality by fine-tuning the PLM on MS-COCO, but limitations in referring to real-world objects remain due to a limited vocabulary (2.05 on the grounded property). Also, see qualitative examples in Fig. 5. Further, we studied the limitation of hallucination in PLM, which may result in irrelevant or incorrect information,

Method	Zero-shot	CLIP-S^{Ref}	CLIP-S	PP
VinVL [57]		**0.83**	0.780	24.16
BLIP [18]		0.82	0.759	27.738
ZeroCap [45]	✓	0.778	0.870	25.737
MAGIC [42]	✓	0.763	0.737	37.126
Ours	✓	0.798	**0.885**	**19.049**

Table 2: Results for image captioning methods. CLIP-S^{Ref} is a supervised metric and the others are not. CLIP-S = CLIP-based image-text similarity, and PP = caption perplexity.

Method	**Hallucinations**		**Human-like**		**Grounded**		**MOS**	
	Images	Videos	Images	Videos	Images	Videos	Images	Videos
MAGIC [42]	2.05	2.06	3.11	3.53	2.05	2.00	1.65	1.77
ZeroCap [45]	3.38	2.46	2.27	2.66	3.66	2.93	2.52	2.30
Ours	**4.44**	**4.53**	**4.00**	**3.66**	**4.66**	**4.26**	**4.01**	**4.14**

Table 3: Hallucinations, Human-like, Grounded and Mean Opinion scores (scale of 1–5) for caption quality using real-world images and videos.

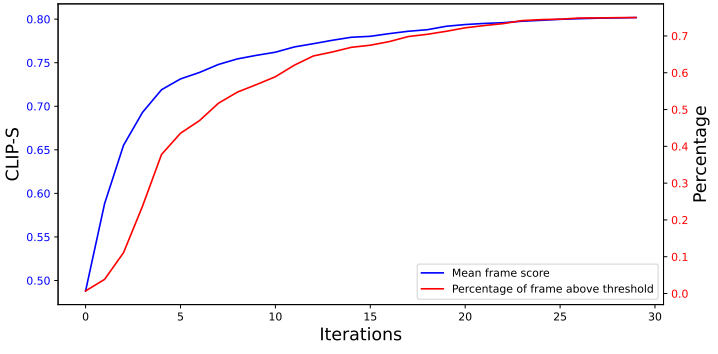


Figure 4: The number of iterations vs. the mean CLIP-S and fraction of frames with a score above 0.75.

our method achieved the best results (4.53 vs. 2.46). We hypothesize that optimizing entire sentences reduces the likelihood of generating random words or irrelevant information (e.g., blockchain). Additionally, fine-tuning the language model as in MAGIC can lead to bias of frequent occurrence of words appearing in MS-COCO captions (e.g., "clock").

Method	B@1	B@2	B@3	B@4	M	R	C	S	CLIP-S	PP
MAGIC [42]	0.160	0.055	0.017	0	0.069	0.147	0.231	0.087	0.766	70.2
ZeroCap [45]	0.127	0.050	0.023	0.010	0.087	0.175	0.471	0.171	0.840	117.4
Ours	**0.254**	**0.082**	**0.029**	**0.011**	**0.147**	**0.208**	**0.527**	**0.183**	**0.892**	**22.7**

Table 4: Results on CLIP-encoded Text inversion. We evaluate metrics that measure text correspondence to the original caption text. Results are reported on other zero-shot captioning methods that employ CLIP.

Additional Quality Studies We evaluate three zero-shot video captioning properties. First, our method, despite being invariant to the order of the frames, maintains a logical

ZeroCap: A genius CEO is not a genius in the world of of Silicon Valley billionaires
MAGIC: A man smiling while holding glasses of wine.
Ours: Microsoft billionaire and philanthropist Bill Gates, who is chairman of the foundation that has been criticized for supporting..

ZeroCap: A wall in the Chinese city of Gansu is a great hit hit.
MAGIC: A view of a big tower with a clock on it.
Ours: The world's largest wall in China, complete with a stunning view from above.

ZeroCap: A city in the Chinese blockchain network Zha dong (not a city in
MAGIC: A view of a city street from a tower.
Ours: Beijing's futuristic office building, which is expected to be one of the most expensive buildings in history.

Figure 5: Examples of our image captions on examples that require real-world knowledge, with two zero-shot image captioning baselines.

order, leveraging the LM and CLIP pretraining information. We find that altering the event order leads to decreased PP and CLIP scores, as demonstrated in Appendix A.1.

Second, we examine whether our method is capable of *describing multiple unrelated frames* with one coherent sentence, see Appendix A.4. Moreover, in Fig. 4, we show that with more iterations, the caption integrates information from multiple frames by measuring the CLIP-S of videos from MSR-VTT dataset. We observe that as iterations progress, the captions describe more frames (i.e., more frames are above the 0.75 CLIP-S threshold (red)). Additionally, Fig. 15 in the appendix illustrates how the caption incorporates information from 4 unrelated images. For more qualitative examples, please refer to Appendix A.7.

Third, we assess the grounding abilities. For this, we measure the ability of our method to act as a *text inversion* method for CLIP's embedding space without considering any image. For this, we encode an image's caption using CLIP and then invert it with either our method or one of the other zero-shot captioning methods. In Tab. 4, we show two insights: (i) for MAGIC, it is difficult to generalize beyond MS-COCO caption styling, and (ii) our inverted text is more fluent than other zero-shot methods (PP score of 22.7 vs. 70.2) and better corresponds to the original caption (CLIP score of 0.89 vs. 0.84). The full experiment details are available in Appendix A.5.

5 Conclusions

We present a method for creating natural-sounding captions from a video. The method is based on learning, for each input, a sequence of vectors that serve as pseudo-tokens that drive the generation process. Once a caption is generated, we update the pseudo-tokens. The process repeats, using the learned pseudo-tokens as the starting point, leading to increasingly concrete and well-grounded captions. Our experiments show that our model generates novel captions that ground objects from multiple images into one coherent narrative.

Acknowledgments This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research, innovation programme (grant ERC CoG 6725974).

References

[1] Ameen Ali, Idan Schwartz, Tamir Hazan, and Lior Wolf. Video and text matching with conditioned embeddings. In *WACV*, 2022.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL*, 2005.

[4] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-based clip-guided essence transfer. *ECCV*, 2022.

[5] Haoran Chen, Jianmin Li, and Xiaolin Hu. Delving deeper into the decoder for video captioning. *ECAI*, 2021.

[6] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *ICLR*, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. *ECCV*, 2020.

[9] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[10] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *EMNLP*, 2020.

[11] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *EMNLP*, 2021.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[13] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Fisher vectors derived from hybrid gaussian-laplacian mixture models for image annotation. *arXiv preprint arXiv:1411.7399*, 2014.

[14] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

[15] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022.

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[17] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. *arXiv preprint arXiv:2112.09583*, 2021.

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. *arXiv e-prints*, art. arXiv:2201.12086, January 2022. doi: 10.48550/arXiv.2201.12086.

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[20] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *ACL*, 2021.

[21] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

[24] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[25] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *CoRR*, abs/1412.6632, 2014.

[26] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021.

[27] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[28] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2001.

[30] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

[31] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[32] Jesus Perez-Martin, Benjamin Bustos, and Jorge Perez. Improving video captioning with temporal composition of a visual-syntactic embedding. In *WACV*, 2021.

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

[35] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[36] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. High-order attention models for visual question answering. *NIPS*, 2017.

[37] Idan Schwartz, Alexander G Schwing, and Tamir Hazan. A simple baseline for audio-visual scene-aware dialog. In *CVPR*, 2019.

[38] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[39] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[40] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *EMNLP*, 2020.

[41] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*, 2014.

[42] Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. Language Models Can See: Plugging Visual Controls in Text Generation. *arXiv e-prints*, art. arXiv:2205.02655, May 2022.

[43] Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. Language models can see: Plugging visual controls in text generation. *arXiv preprint arXiv:2205.02655*, 2022.

[44] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *MM*, 2021.

[45] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *CVPR*, 2022.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[47] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2014.

[48] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. *EMNLP*, 2019.

[49] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep Learning for Video Classification and Captioning. *arXiv e-prints*, art. arXiv:1609.06782, September 2016.

[50] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019.

[51] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *EMNLP*, 2021.

[52] Jun Xu, Tao Mei, Ting Yao, and Yong 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*, pages 5288–5296, 2016.

[53] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[54] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019.

[55] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.

[56] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[57] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, 2021.

[58] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. *arXiv preprint arXiv:1711.08496*, 2017.

[59] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.