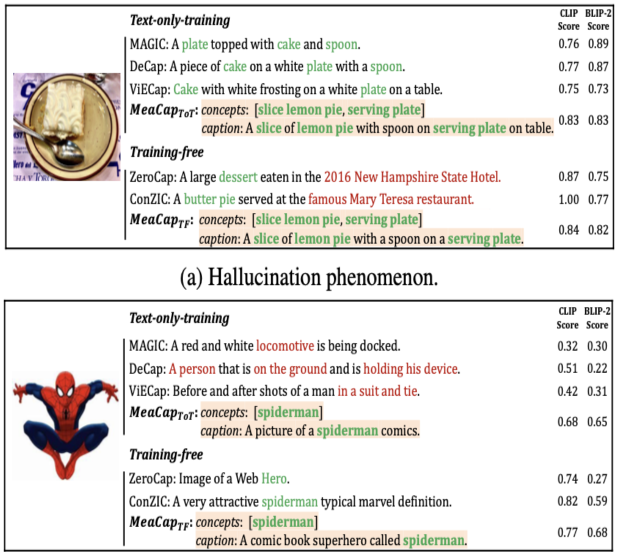
CSE 597: Vision and Language Course Project Report

MeaCap: Memory-Augmented Zero Shot Image Captioning

Ronit Bhansali
rxb5803@psu.edu



Text-only-training	CLIP Score	BLIP-2 Score
MAGIC: A plate topped with cake and spoon.	0.76	0.89
DeCap: A piece of cake on a white plate with a spoon.	0.77	0.87
ViECap: Cake with white frosting on a white plate on a table.	0.75	0.73
MeaCap$_{ToT}$: *concepts*: [**slice lemon pie**, **serving plate**] *caption*: A **slice** of **lemon pie** with spoon on **serving plate** on table.	0.83	0.83
Training-free		
ZeroCap: A large dessert eaten in the 2016 New Hampshire State Hotel.	0.87	0.75
ConZIC: A butter pie served at the famous Mary Teresa restaurant.	1.00	0.77
MeaCap$_{TF}$: *concepts*: [**slice lemon pie**, **serving plate**] *caption*: A **slice** of **lemon pie** with a spoon on a **serving plate**.	0.84	0.82

(a) Hallucination phenomenon.

Text-only-training	CLIP Score	BLIP-2 Score
MAGIC: A red and white locomotive is being docked.	0.32	0.30
DeCap: A person that is on the ground and is holding his device.	0.51	0.22
ViECap: Before and after shots of a man in a suit and tie.	0.42	0.31
MeaCap$_{ToT}$: *concepts*: [**spiderman**] *caption*: A picture of a **spiderman** comics.	0.68	0.65
Training-free		
ZeroCap: Image of a Web Hero.	0.74	0.27
ConZIC: A very attractive spiderman typical marvel definition.	0.82	0.59
MeaCap$_{TF}$: *concepts*: [**spiderman**] *caption*: A comic book superhero called **spiderman**.	0.77	0.68

(b) Image contains world knowledge.

Figure 1. This figure is directly used from the MeaCap paper. The proposed MeaCap method where the red is incorrect and green is correct. In (a) we notice that training free methods associate the *pie* with incorrect location information, which actually get high marks in CLIP score. This might be due to the fact that CLIP is trained on web-scale noisy image-text data. In (b) we notice that existing text only training methods fail to generate *spiderman* as some training free methods do, but the proposed text only training method of MeaCap$_{\mathrm{ToT}}$ can do that effectively.

1. Task

The state-of-the-art (SOTA) method used for tackling zero shot image captioning in this paper is MeaCap. The idea of zero shot image captioning is to generate captions for images without relying on well paired image and text data. The main challenges that are faced while implementing this is that training free methods may hallucinate unrelated details and text only training methods often lose their generalization capabilities.

2.1 Related Work

Several papers preceding MeaCap have tried to address zero shot image captioning however they have certain flaws which MeaCap tackles better, thereby resulting in a performance improvement across multiple metrics.

1. **ZeroCap [1]**: ZeroCap is utilizing pre-trained vision and language (e.g. GPT-2) for zero shot image caption without any fine tuning. It iteratively generates captions by optimizing image-text similarity metrics. Although it promises strong generalization ZeroCap often hallucinates and produces imaginary results which are not present in the image, this limits the practical applicability of ZeroCap.

2. **MAGIC [2]**: MAGIC follows a fine-tuning methodology using high quality textual data to train language models under CLIP supervision. It regularizes the generation process with the CLIP induced scores leading to meaningful captions aligned with the image. The fine-tuning causes a loss of pre-trained knowledge leading to bad performance in out of domain images.

3. **DeCap [3]**: DeCap explores a memory-augmented retrieval framework where the captions are retrieved from the textual memory using CLIP embeddings and then used to guide generation. Although it generates relevant captions it faces the same problem as MAGIC, which is cross domain evaluation, it shows decreased performance in images which differ from those in the training set.

4. **ConZIC [4]**: ConZIC implements Gibbs sampling along with a non-autoregressive language model with zero-shot captioning. Although this approach enhances diversity and inference speed, but it struggles to maintain coherence and accuracy for complex images.

5. **ViECap [5]**: ViECap introduces a method for transferring visual entities into text generation, this approach leverages transferable decoding for enhanced zero shot. It underperforms on datasets with abstract or uncommon visual features due to limited diversity in the training set.

2.2 State-of-the-Art (SOTA) Method

The State-of-the-Art method for tackling zero shot image captioning is MeaCap, unlike previous methods it incorporates the following:

1) A *retrieve then filter module* which is responsible for extracting highly relevant visual concepts from the external textual memory. This process reduces hallucinations and improves focus on actual image content.

2) A *memory augmented visual related fusion core* which integrates image-text and text-text similarity making sure that the image and the caption being generated are closely related.

3) The use of *CBART language model* which refines the captions iteratively leveraging the extracted key concepts for fluency and coherence.

MeaCap's dual capability of training free and text only mode allow it the versatility to be used for a variety of tasks and use cases while still performing well. It outperforms previous models on benchmark datasets like MSCOCO and NoCaps achieving better numbers of metrics like BLEU, SPICE and CIDEr. It's ability to leverage augmented memory for consistent performance for in-domain and out-of-domain images make it the current State-of-the-Art method for zero shot image captioning.

3.1 Approach

The MeaCap framework consists of three primary components which are designed to overcome the challenges of zero shot image captioning.

1) *Retrieve-then-Filter Module*: This model retrieves relevant image captions from a pre-constructed textual memory to filter out redundant information. The memory is built on a large corpus like the CC3M dataset, and the sentences are encoded using the CLIP text encoder. Cosine similarity is also used to compare the similarity of an image's embedding with the textual memory embedding. The retrieved captions are then parsed into subject-predicate-object triplets and similar concepts are clustered using the sentence-BERT embeddings. Clusters are subsequently filled out based on appearance frequency and relevance.

2) *Memory-Augmented*: For guiding caption generation MeaCap introduces a fusion score combining which uses the Fluency score () generated by a pre-trained language model. The cosine similarity is also computed between the image and text using CLIP. The text in text in text model similarity () evaluates the generated captions and memory concepts. The final fusion score is a weighted sum of all of them.

$$p^{fusion} = \alpha p^{lm} + \beta p^{ITs} + \gamma p^{TTs}$$

3) *Keyword-to-sentence CBART*: MeaCap refines the generated caption iteratively using CBART, it keeps the key concepts as lexical constraints and performs replacement (replacing a token with a more accurate one), Insertion (adds missing tokens to improve fluency and coherence) and Termination (stops when the caption achieves full coherence and fluency). It performs these steps iteratively.

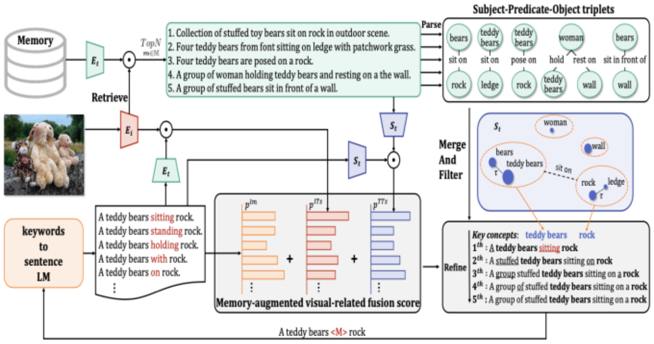

Subject-Predicate-Object triplets
Memory
E_t
TopN
m∈M
1. Collection of stuffed toy bears sit on rock in outdoor scene.
2. Four teddy bears from font sitting on ledge with patchwork grass.
3. Four teddy bears are posed on a rock.
4. A group of woman holding teddy bears and resting on a the wall.
5. A group of stuffed bears sit in front of a wall.
Parse
bears
teddy bears
teddy bears
woman
bears
sit on
sit on
pose on
hold
rest on
sit in front of
rock
ledge
rock
teddy bears
wall
wall
Retrieve
E_i
E_t
S_t
S_t
Merge And Filter
S_t
woman
wall
bears
teddy bears
sit on
rock
ledge
τ
τ
keywords to sentence LM
A teddy bears sitting rock.
A teddy bears standing rock.
A teddy bears holding rock.
A teddy bears with rock.
A teddy bears on rock.
p^{lm}
p^{ITs}
p^{TTs}
+
+
Memory-augmented visual-related fusion score
Refine
Key concepts: teddy bears rock
1th : A teddy bears sitting rock
2th : A stuffed teddy bears sitting on rock
3th : A group stuffed teddy bears sitting on a rock
4th : A group of stuffed teddy bears sitting on a rock
5th : A group of stuffed teddy bears sitting on a rock
A teddy bears <M> rock

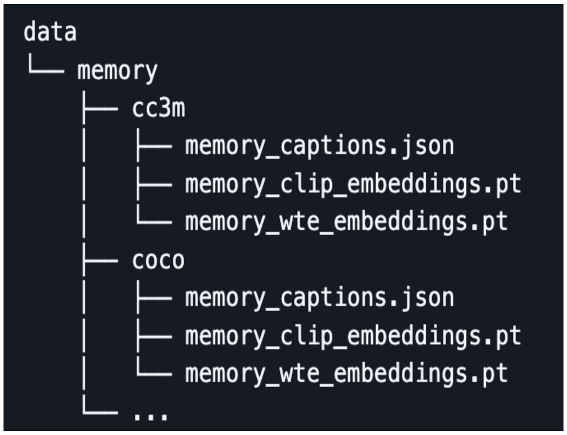
Figure 2. This figure is directly used from the MeaCap paper. This figure illustrates the data flow in MeaCap showing retrieval, filtering, fusion score and caption generation scoring.

3.2 My Implementation

1) I implemented the model in google colab and the first step is to download all the prerequisites like torch vision, CLIP, Bart, networkx, sentence transformer and so forth. The next step is to clone https://github.com/joeyz0z/MeaCap and mount the google drive. Install python 3.9 and the required libraries.

2) The next step is to download the captions from the GitHub link CC3M, SS1M, COCO & Flickr30k and create the preprocessed textual memory.

3) The file structure must be made in a certain way to ensure smooth operation of the project. These preprocessed memory files should be placed in./data/memory, modify path as per your needs.



```
data
└── memory
    ├── cc3m
    │   ├── memory_captions.json
    │   ├── memory_clip_embeddings.pt
    │   └── memory_wte_embeddings.pt
    ├── coco
    │   ├── memory_captions.json
    │   ├── memory_clip_embeddings.pt
    │   └── memory_wte_embeddings.pt
    └── ...
```

Figure 3. This figure is directly used from the MeaCap GitHub. This figure illustrates the file organization of the pre-processed model captions.

4) Download the OpenAI CLIP Vision Transformer model which is responsible for extracting the image content. Download the scene graph parser to understand object relationships and finally sentence-BERT to relate context in different sentences. I Updated the paths accordingly.

5) After installing and setting up the project requirements and datasets accordingly I tested the implementation for training free MeaCap using the following command: *python inference.py --use_prompt --memory_id cc3m --img_path ./image_example --lm_model_path ./checkpoints/CBART_one_billion*. I proceeded to also test the text only MeaCap with the following command: *python inference.py --memory_id coco --img_path ./image_example --lm_model_path ./checkpoints/CBART_COCO*

6) I proceeded further after confirming the proper working of text only and training free MeaCap especially text only MeaCap since it is the true SOTA method. I downloaded the Flickr30K dataset along with the annotations from the official link. I further processed Flickr30k using the Karpathy split for training, testing and validation.

7) I created a script to generate evaluation metrics like SPICE, CIDEr, BLEU and METEOR and used the cocoeval.py API to compute the metrics using the following command: python cocoeval.py --generated_json ./outputs --ground_truth ./annotations/flickr30k.json

3.3 Reused Code and Custom Code

- The code and libraries I have re-used is from PyTorch, Sentence-Transformers, Flickr30K, Hugging Face, CLIP, SceneGraphParser, Networkx. I also downloaded different models from hugging face to test results

- I created the main.Ipynb to run the final code which executed the final zero shot implementation on the Flickr30k and presented the final results. I also used Cocoeval to evaluate metrics like CIDEr, BLEU, SPICE and METEOR. I extracted the images from the Flickr30k data set using the Karpathy split. I have written scripts for all of the above tasks. Formatting the Json is also required to evaluate results properly.

4. Dataset

- I primarily used the Flickr30k dataset which involves 31k images, the Karpathy split suggests that the training split contain 29k images, testing split contain 1k images and the validation split contain 1k images as well. The Flickr30k dataset has 5 captions for each image describing the people and activities in the pictures.

- I also used the COCO (Common objects in context) dataset to generate the textual memory of captions which has a corpus of 330,000 annotated images with multiple captions for each image. The paper also declared the use the SS1M and CC3M datasets however I have not used them in my replication.

- For pre-processing I have resized all images to 224x224 pixels and normalized them using the CLIP model's preprocessing pipeline. The text is preprocessed using sentence-BERT and CLIP's text tokenizer which generates subject-predicate-object triplets which are further used for filtering.

5.1 Result Metrics

- *BLEU (Bilingual Evaluation Understudy)*: This metric measures the overlap between the n-grams of a generated caption and reference caption. It uses precision-based scoring thereby focusing on the proportion of matching n-grams between the generated caption and the reference caption. BLEU-1 to BLEU-4 indicates 1-gram to 4-gram matching respectively.

- *CIDEr (Consensus Based Image Description Evaluation)*: This metric measures the similarity between the generated caption and the reference caption focusing on n-grams of overlap.

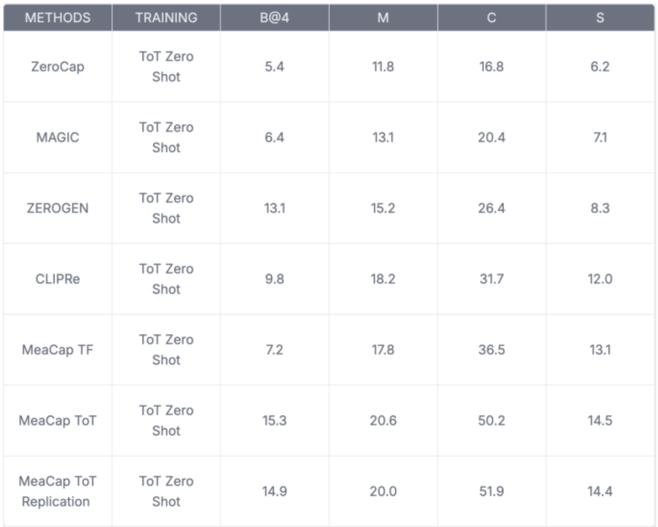
- *METEOR (Metric for Evaluation of Translation without Ordering)*: This evaluation approach considers unigram matches considering stemming, synonyms and paraphrases. It also calculates a harmonic mean of precision and recall of the unigrams between the generated captions and reference captions while prioritizing the recall with a higher weight.

- *SPICE (Semantic Propositional Image Caption Evaluation)*: Focuses on semantic correctness by comparing scene graphs of the generated captions and the reference captions because scene graphs inherently support semantic richness. It also measures the ability to identify objects, relationships and attributes correctly.

- *ROUGE (Recall oriented Understudy for Gisting Evaluation): Measures the overlap between n-grams, word sequences or word pairs between the generated captions and reference captions. More emphasis is given to recall. ROUGE-N measures the overlap of n-grams. ROUGE-L measures the longest common subsequence between the reference and generated caption. ROUGE-W is a weighted variant of ROUGE-L emphasizing on consecutive sequences.*

5.1 Quantitative Results

The table shows comparative results on various metrics which are relevant for Image Captioning like BLEU, CIDEr, SPICE, METEOR on pre-existing or baseline training methods and the MeCap Training method. The MeCap TF stands for MeCap Training Free and MeCap ToT stands for MeCap Text only Training. MeCap ToT Replication is the model I have replicated and achieved similar results with.

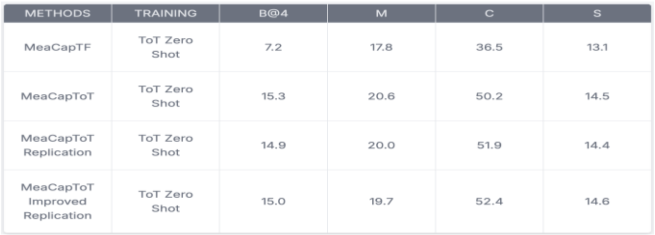


METHODS	TRAINING	B@4	M	C	S
ZeroCap	ToT Zero Shot	5.4	11.8	16.8	6.2
MAGIC	ToT Zero Shot	6.4	13.1	20.4	7.1
ZEROGEN	ToT Zero Shot	13.1	15.2	26.4	8.3
CLIPRe	ToT Zero Shot	9.8	18.2	31.7	12.0
MeaCap TF	ToT Zero Shot	7.2	17.8	36.5	13.1
MeaCap ToT	ToT Zero Shot	15.3	20.6	50.2	14.5
MeaCap ToT Replication	ToT Zero Shot	14.9	20.0	51.9	14.4

Figure 4. This figure represents the scores on the Flickr30k dataset of MeCap ToT Replication versus the original MeCap implementation and baseline models like ZeroCap, MAGIC, etc. B@4 stands for BLEU-4, M stands for METEOR, C stands for CIDEr, and S stands for SPICE. The similarity Threshold for Concept filtering is set at 0.55 in my replication which is in coherence with the value being used in the paper. It is the most optimal value since it retains the most relevant concepts and filters out the noisy ones.

6. Possible Improvements and Results

I identified that the current similarity cosine threshold for concept filtering of 0.55 is static and a dynamic similarity threshold may work better. I implemented the dynamic similarity based on dataset characteristics for e.g. average similarity scores or density of relevant concepts. Current clustering only considers embedding similarity between concepts while ignoring the context of the relationship (e.g. 'bear on a rock' vs 'rock in a park'). Therefore, I implemented contextual clustering that incorporates relationships (subject-predicate-object triplets) into the clustering process to group more semantically meaningful concepts. I used the spectral clustering algorithm to achieve this. I saw slight improvement upon implementation of these ideas.



METHODS	TRAINING	B@4	M	C	S
MeaCapTF	ToT Zero Shot	7.2	17.8	36.5	13.1
MeaCapToT	ToT Zero Shot	15.3	20.6	50.2	14.5
MeaCapToT Replication	ToT Zero Shot	14.9	20.0	51.9	14.4
MeaCapToT Improved Replication	ToT Zero Shot	15.0	19.7	52.4	14.6

Figure 5. This figure represents the scores on the Flickr30k dataset of MeCap ToT Improved Replication versus the MeCap Replication and MeCap Original (Training Free and Text only Training) implementation. B@4 stands for BLEU-4, M stands for METEOR, C stands for CIDEr, and S stands for SPICE.

7. Code Repository

You can find my public MeaCap Replication repository in the link provided below.

https://github.com/Ronit26x/MeaCapReplication

or

link

References

1) Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022.
2) 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 gen- eration. *arXiv preprint arXiv:2205.02655*, 2022.
3) Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding clip latents for zero-shot captioning via text-only training. *arXiv preprint arXiv:2303.03032*, 2023.
4) Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retriev- ing from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.
5) Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable decoding with visual entities for zero-shot image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3136–3146, 2023.
6) Radford, A., Kim, J. W., Hallacy, C., et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision.* Proceedings of the International Conference on Machine Learning (ICML).
7) Yang, J., Lu, J., Lee, S., et al. (2018). *Graph R-CNN for Scene Graph Generation.* Proceedings of the European Conference on Computer Vision (ECCV).
8) Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). *BLEU: a Method for Automatic Evaluation of Machine Translation.* Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL).
9) Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). *SPICE: Semantic Propositional Image Caption Evaluation.* Proceedings of the European Conference on Computer Vision (ECCV).
10) Rennie, S. J., Marcheret, E., Mroueh, Y., et al. (2017). *Self-Critical Sequence Training for Image Captioning.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
11) Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y**.** (2019). *From Recognition to Cognition: Visual Commonsense Reasoning.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
12) Kim, T., Ahn, P., Kim, S., et al. (2023). *NICE: CVPR 2023 Challenge on Zero-shot Image Captioning***.** Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
13) Qi, J., Xu, Z., Shao, R., et al. (2024). *RoRA-VLM: Robust Retrieval-Augmented Vision Language Models.* arXiv preprint arXiv:2410.08876**.**
14) Hu, Z., Iscen, A., Sun, C., et al**.** (2022). *REVEAL: Retrieval-Augmented Visual-Language Pre-Training with Multi-Source Multimodal Knowledge Memory.* arXiv preprint arXiv:2212.05221.