CSE-597: Vision & Language - Course project report

MeaCap: Memory-Augmented Zero-shot Image Captioning

Vedant Sawant
vzs5407@psu.edu

1. ABSTRACT

Memory-Augmented zero-shot image Captioning framework is a novel method equipped with textual memory which will retrieve important key concepts which are related to the image and then filter this information required accordingly. MeaCap can generate concept-centered captions that keep high consistency with the image with fewer hallucinations and more world-knowledge. I have replicated this project and tried to improve results with different experiments.

2. TASK

Image captioning is understanding the image and generating text which explains the image in question. They have addressed the issue in Zero Shot method for image Captioning. Zero Shot learning is the ability of the model to perform its intended task without having specific training on the dataset they are supposed to be tested on. The generalization of the model during its training helps the model to generate correct predictions or decisions. The most well-known Zero shot example we know is LLM by OpenAI's ChatGPT which can answer any question without being tested on the dataset.

2.1 Challenges

2.1.1 Dependability on Pretrained knowledge

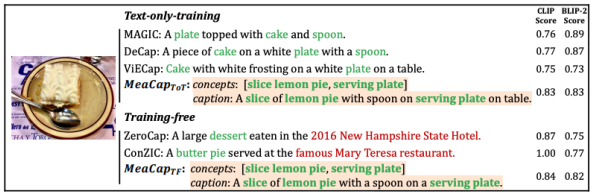
As stated in definition the model depends too much on the pretrained knowledge. If any task arrives where no relevant knowledge is available it will not able to answer it correctly.

2.1.2 High Probability of misinterpretation

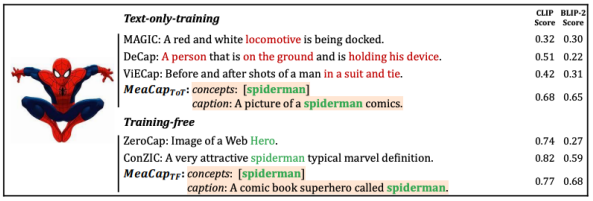
A question asked may share some weak association with knowledge it trains on then the model will tend to hallucinate towards the weak association leading to errors.

2.1.3 Excessive Generalization

Model may tend to generalize new task which for its domain and will fail to identify small differences which require domain knowledge



(a) Hallucination phenomenon.



(b) Image contains world knowledge.

Figure 1. Taken from research paperThe motivation for MeaCap where the red is incorrect and green is correct. (a) 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. (b) Existing text-only-training (ToT) methods fail to generate spiderman as some training-free methods do, but the ToT version of our method (MeaCapToT) can also do that.

2.2 Advantages

2.2.1 No Dataset needed

This method eliminates the need for dataset to be labelled by a user which is very cost expensive and time consuming.

2.2.2 Adaptability

The model can easily adapt to tasks which are in various domains while other supervised models would require time to adapt to the tasks.

2.2.3 Less Computation Cost

We don't need to retrain the model for every new task we define to use a model. We can directly use a Zero shot model and optimize it according to task.

2.3 Ideal Scenarios to use Zero Shot

1) When there is no labelled dataset available for the task or when you are on your initial stages of collecting the dataset.

2) When task is very dynamic and requirements is changing rapidly.
3) When you are in lack of time or resources to implement the task.

2.4 Scenarios to avoid using Zero Shot

1) When high accuracy is very crucial for the task to be done. Domain like health-care, defense, weapons industry where accuracy is very critical.
2) When computation resources are not a problem while executing a task.
3) When domain specific knowledge is very important. For example: Traffic CCTV capturing license plate using object detection.

3. Related Work

3.1 Supervised Captioning

Supervised Captioning uses image-text pairs and trains on them. Early models used to construct CNN for extracting features from the image and then uses LSTM-RNN based network to generate the captions for it. Some of the paper even used object detection to extract the region of interest with help of which you can identify the captions.

3.2 Zero shot image captioning.

Recently, zero-shot IC has gained more and more attention, which targets at generating image captions under two cases: i) without any data for training named training-free zeroshot IC; ii) just using text from the captioning dataset to train the LM named text-only-training zero-shot IC.
Training-free methods realizes the zero-shot IC via the pre-trained vision-language model [45], to guide the generation of a pre-trained LM.
Text-only-training methods train or fine-tune the text decoder just using the corpus from the captioning dataset.

3.3 External memory in image captioning

External memory is proven to be useful in various vision and language tasks. SmallCAP is an example which uses CLIP to retrieve relevant captions and then takes caption as a prompt such that prompt helps the LLM to generate the captions accurately rather than blindly using the model to get the caption.

3.4 ConZIC(Controllable Zero-shot Image Caption)

[1]This compiles Gibbs sampling with non-auto-regressive language models to enhance diversity and speed of zero shot captioning.

3.5 MAGIC(Multimodal Augmented Generation for Image Captioning)[2]

It uses text data fed to it to generate captions. This issue with this method is that it lacks generalization as it might have an issue to caption image outside its data.

3.6 ZeroCap: Zero-Shot Image-to-Text Generation[3]

ZeroCap uses pre-trained vision-language models, such as CLIP, to help language models in creating captions without additional training. It has a high rate of hallucinations because of no descriptions not present.

3.7 State of the art (SOTA) method for tackling the particular task.

The method "MeaCap: Memory-Augmented Zero-shot Image Captioning" is a state of art because it achieves superior performance than other SOTA considered methods. This is considered SOTA because of following reasons.
1) Ability to handle hallucination
2) Use external memory to get deep knowledge hence can incorporate world knowledge.
3) Uses high quality LLM like CBART_ONE_BILLION to produce fluent sentences.

4. Approach

4.1 Proposed Method

Step 1.
Extract the features from the image in question of which captions need to be done. They proposed to use pretrained computer vision CLIP model (clip-vit-base-patch32).
STEP 2.
From the extracted visual features search the external textual memory get captions which are closely related to the image content.
STEP 3.
Find Visual Related Fusion Score which is by aligning image content with text. Top aligned scores are considered and passed further to processing.
STEP 4.
After getting all the text with high scores it uses Text2Text Generation model to complete the sentence and get the caption.

4.2 My Implementation

1) First, I cloned the git repository from the GitHub link provided in the paper and uploaded everything to google drive. I used to google colab to implement the code for the project. (https://github.com/joeyz0z/MeaCap)
2) Next, I created a virtual environment of version python3.9. I checked the version requirements and no version requirements were given. I initially tested with

3.11 and I got an issue in execution of code so then I installed python version 3.9 and it ran successfully.
3) Next step was downloading the external memory for image captioning. Three links are provided in the GitHub repository.

CC3M: Link
SS1M: Link
COCO: Link
Flickr30k: Link
These contains all the captions in the dataset so that while generating caption the proposed method can refer to this stored caption.
Put these files in data folder such that the folder structure look like image below.

```
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
    └── ...
```

Fig2. Taken from github shows the file structure of data folder

If you already have a memory captions.json then create a memory embeddings with below command:

```
python prepare_embedding.py --memory_id coco --memory_path data/memory/coco/memory_captions.json
```

Adjust your path accordingly.
4) Download the openai/clip-vit-base-patch32 model which extracts the image content. Click here to download.
5) Download the SceneGraphParser lizhuang144/flan-t5-base-VG-factual-sg which helps us understand the objects and relationship. Click here to download.
6) Download the SentenceBERT which helps in connecting different context into sentence. Click here to download.
Modify paths of all these downloaded files in get_args file.
7) Install requirements.txt and also install sentence_transformer as it is not mentioned in the requirements.
8) Then I received an error in the sentence_transformer class where it was unable to find cache_download. We need to backspace it because of some compatible issue with library.
9) Then test it by executing 3 sets of command which will tell us whether captioning is working or not.
First is Training-free:

```
python inference.py --use_prompt
--memory_id cc3m --img_path
./image_example --lm_model_path
./checkpoints/CBART_one_billion
```

Second is Text-only-training:

```
python inference.py --memory_id coco
--img_path ./image_example --
lm_model_path ./checkpoints/CBART_COCO
```

Third is Memory_Concepts + Viecap:

```
python viecap_inference.py --memory_id
coco --image_path "*.jpg" --weight_path
"checkpoints/train_coco/coco_prefix-
0014.pt"
```

10) This will create captions of file in ./image_examples.
11) From checking the results given in the paper and comparison with other SOTA models. MeaCap$_{TOT}$ which is Text-only-Training with MeaCap method where memory_id is coco and CBART_COCO as language model. So, I implemented only MeaCap$_{TOT}$ improved it.
12) After this I downloaded Flickr30k Dataset with images and its annotations.
13) I used Karpathy split to split the dataset in training, testing and validation and put all testing images in image_examples folder.
14) Make changes in code such that the json file of output you get is given below in the image.


```
[
  {
    "image_id": "7253781238",
    "caption": "A young woman is riding a skate board in the public park area."
  },
  {
    "image_id": "725722798",
    "caption": "A upside down skateboard player on the ground in action."
  },
  {
    "image_id": "7258153438",
    "caption": "Two people wearing helmets and talking on cell phones."
  },
  {
    "image_id": "7258326556",
    "caption": "A man sampling several barrels of wine into a glass."
  },
```


Fig3. Taken my own file which shows format of json

15) The I run text-only-training command on all images in image and we get the json of the output in ./outputs folder.
16) Get the flickr30k annotation and edit it such that you file looks like image below.


```
{
 "images": [
   {"id": "134206"},
   {"id": "148284"},
   {"id": "36979"},
   {"id": "65567"},
   {"id": "81641"},
   {"id": "178045"}
 ],
 "annotations": [
   {"id": "134206", "image_id": "134206", "caption": "A baseball player fields during a game of professional ball."},
   {"id": "148284", "image_id": "148284", "caption": "A man traveling down the counter carrying a multi-colored suitcase."},
   {"id": "36979", "image_id": "36979", "caption": "Poker players enjoying wine at table with game."},
   {"id": "65567", "image_id": "65567", "caption": "A man and red beautiful young woman hold hands beside a bride in an elegant suit and tie."},
   {"id": "81641", "image_id": "81641", "caption": "A windtower of Iran legacies mosque building."},
   {"id": "178045", "image_id": "178045", "caption": "A woman wearing an orange dress is riding a white horse in the equestrian ring."}
 ]
}
```

Fig4. Taken my own file which shows format of json

17) Once you have both json next step is calculating metrics. The metrics used to evaluate model are BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE_L, CIDEr, SPICE
18) This metric was not implemented in this repository. So, I created a new file for metric calculation. Run cocoeval.py to get the metrics.

4.3 Libraries

sentence_transformer, hugging_face, nltk, numpy, Pillow, torch, torchvision, torchaudio, psutil, sacremose, transformer.

4.4 Additional Code

I have written codes by myself:
1) Cocoeval for metric evaluation of 2 json
2) Creating json according to the requirements
3) Download different huggingface models,
4) Ipynb to execute whole code,
5) Changed the current logic of creating the results json
6) Changed code inside the repo in case of using 2 different model it gives an error for size of tensor not matched.
7) Creating Karpathy split dataset.

4.5 Reused

1) openai/clip-vit-base-patch32
2) lizhuang144/flan-t5-base-VG-factual-sg
3) sentence-transformers/all-MiniLM-L6-v2
4) COCO captions
5) Flicker30k Dataset
6) CBART_COCO LM model

5. Dataset

- Flickr30k: A dataset of 30,000 images with multiple captions per image. I have used this dataset for comparing the result of improved method with the original proposed. I used Karpathy split on this data set such that testing set contained 1000 images, validation contains 1000 and training contains 29000 images. This state of split of images remains the same such that all metrics are calculated over same set of data.
- COCO: The Common Objects in Context dataset with images and their corresponding captions. This dataset is used for using the external memory captions.
- Data Preprocessing: After generation of text embeddings and image embeddings we resize the tensor to the smaller. This can happen because of improper size of match of both because different models are used for both the task.

6. Results

6.1 Metrics

- **BLEU (Bilingual Evaluation Understudy)**: BLEU measures the overlap of n-grams between a generated text and a reference text.
BLEU-1 evaluates 1-gram (individual words).
BLEU-2 evaluates 2-grams (word pairs).
BLEU-3 and BLEU-4 extend to trigrams and 4-grams, respectively. Higher BLEU scores indicate closer similarity to the reference.
- **METEOR (Metric for Evaluation of Translation with Explicit ORdering):** Focuses on both precision and recall of word matches between generated and reference texts.
- **CIDEr (Consensus-based Image Description Evaluation):** Measures the similarity of the generated text to multiple reference texts, emphasizing the importance of n-gram overlap.
-**SPICE (Semantic Propositional Image Caption Evaluation):** Focuses on semantic similarity rather than n-gram overlap. Check how well text which is generated captures meaningful relationship.
I compared the values of above given results. Higher the score better the performance of the model.

Methods	Training	B@4 (Flickr30K)	M (Flickr30K)	C (Flickr30K)	S (Flickr30K)
ZeroCap,Ä†	Text-only-training, zero-shot inference	5.4	11.8	16.8	6.2
MAGIC	Text-only-training, zero-shot inference	6.4	13.1	20.4	7.1
ZEROGEN	Text-only-training, zero-shot inference	13.1	15.2	26.4	8.3
CLIPRe	Text-only-training, zero-shot inference	9.8	18.2	31.7	12
MeaCap_TF	Text-only-training, zero-shot inference	7.2	17.8	36.5	13.1
MeaCap_ToT (SOTA) * **	Text-only-training, zero-shot inference	15.3	16.5	30.9	10.3

Fig 6. This figure compares the accuracy metrics of my implemented results with other baseline models.

The above image shows that the results I implemented are better than the baseline models for Zero-Shot Image Captioning.

Metric	My Implementation	MeaCapToT
BLEU	15.3	15.3
METEOR	16.5	20.6
CIDEr	30.9	50.2
SPICE	10.3	14.5

Fig 7. My implementation showed less values than the one replicated in the paper

The above image shows that the implementation that I made has less accuracy metric than compared to what is given in the paper. This could be because i) the paper has not shared the code for the evaluation of the metrics. Ii) Sometimes the variation in causes GPU is slow so variation occurs iii) Different version of libraries because of different configuration of CPU.
iv) Inconsistency and randomness in evaluation pipeline.

7. Possible Improvements and Results

7.1 Improvements

1) I used all-mpnet-base-v2 for sentence transformer as it is designed to generate high quality token and some cases is better than all-MiniLM-L6-v2.
2) Resized the embeddings which are generated for image and text to same size to increase the compatibility of the different models for image and text.
3) Improved the caption size it is considering. Initially it was 4 now it is 8 so that it will include more caption and help to understand the context of the image.

Metric	Previous	Current	Change
BLEU-1	0.542	0.564	+0.022
BLEU-2	0.326	0.344	+0.018
BLEU-3	0.189	0.204	+0.015
BLEU-4	0.105	0.118	+0.013
METEOR	0.165	0.177	+0.012
ROUGE_L	0.370	0.386	+0.016
CIDEr	0.309	0.361	+0.052
SPICE	0.103	0.103	No Change

Fig8. Results after I made the improvements to the code.

8. Code Repository

GitHub link for my project: Click here to access

References

1. Y. Zeng, Q. Huang, and J. He, "MeaCap: Memory-Augmented Zero-shot Image Captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp.
2. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3128–3137, 2015.
3. Zequn Zeng, Hao Zhang, Ruiying Lu, Dongsheng Wang, Bo Chen, and Zhengjue Wang. Conzic: Controllable zero-shot.
4. 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.
5. Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visualsemantic arithmetic.
6. Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2:67–78, 2014.

7. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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.
8. Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pages 65–72, 2005.
9. Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4566–4575, 2015.
10. 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.
11. Rita Ramos, Bruno Martins, Desmond Elliott, and Yova Kementchedjhieva. Smallcap: lightweight image captioning prompted with retrieval augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2840–2849, 2023