Introduction

Method 1: VLM-based Image Retrieval

This method uses a Vision-Language Model (VLM), such as the LLaVA model, to generate descriptions for each image. First, it reads the images and sends them to the model to generate an object list and a short caption of the image. Then, the dialogue text is pre-processed and encoded into embedding vectors by sentence transformer, which are compared with the image description embeddings. Finally, by calculating the cosine similarity of the embedding, the top 30 image with the highest similarity to the dialogue content is selected.

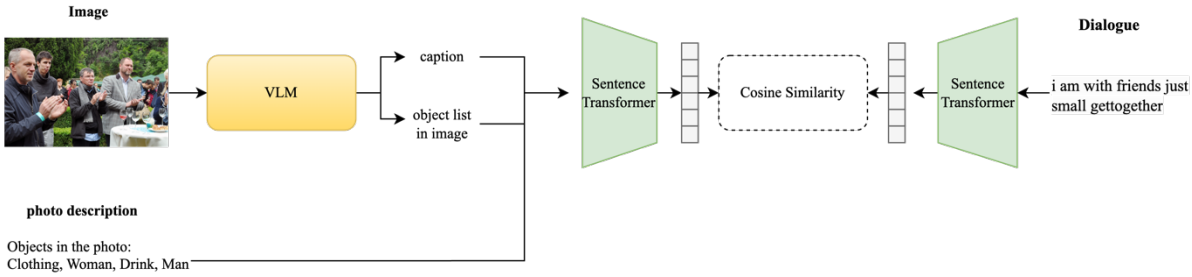

Fig 1: Overview of VLM-based image retrieval method

Method 2: CLIP-based Image Retrieval

This method uses the CLIP model, which maps both text and images into the same embedding space. First pre-processes the dialogue text to extract information relevant to the image. Then, it uses the CLIP model to process the images and the text, obtaining their embedding vectors. Afterward, the similarity between the text and each image is calculated, and the top 30 images with the highest similarity are selected.

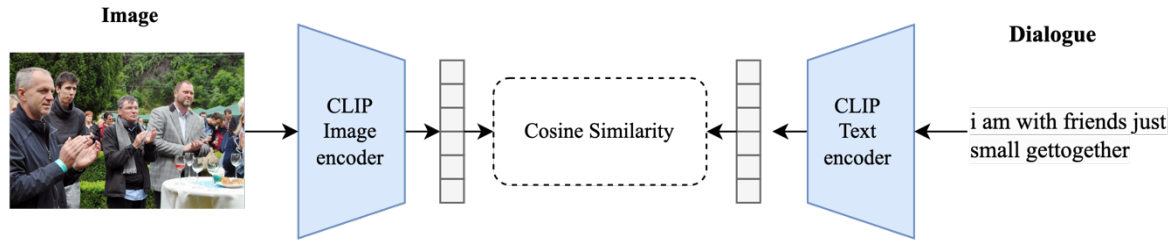

Fig 2: Overview of CLIP-based image retrieval method

Experiment Setting

The VLM-based method uses a Vision-Language Model (e.g., LLaVA 34B) to generate photo descriptions, captions, and object lists for each image. Dialogue text is pre-processed and embedded, then compared with image text embeddings using cosine similarity. The top 30 images with the highest similarity scores are retrieved, evaluated using Recall@30.

The CLIP-based method maps images and dialogue text into a shared embedding space using the CLIP model (clip-vit-large-patch14). Images and pre-processed text are encoded separately, and their embeddings are compared using cosine similarity. The top 30 images are retrieved based on similarity for Recall@30 evaluation.

Both methods used the same dataset and pre-processing for dialogue text. Experiments were conducted in single GPU RTX 3090 environment, and Recall@30 was used as the performance metric to ensure fair comparison.

Experiment Result

Table 1. Compare the two image retrieval methods in test dataset

Method	Recall@30
VLM-based	0.736
CLIP-based	0.690

Table 2. Compare using different input for VLM-based image retrieval method in test dataset

Type of text input	Recall@30
Photo description	0.693
VLM caption	0.526
VLM object list	0.612
Photo description + VLM caption	0.723
Photo description + VLM caption + VLM object list	0.736

Questions

1. What kind of pre-processing did you apply to the photo or dialogue text?

Photo Pre-processing:

- **For Method 1 (VLM-based retrieval)**: The images are read and processed in batches. There is no specific pre-processing applied to the photos themselves before being passed to the image captioning model. The key processing here is related to generating captions for the images by sending them through the `LLaVA` model.
- **For Method 2 (CLIP-based retrieval)**: Each image is loaded and converted to RGB format. The images are then processed using the `CLIPProcessor`, which ensures the images are in the correct format and size for the CLIP model.

Dialogue Text Pre-processing:

- **Common Steps**: For both methods, dialogue text goes through basic pre-processing to prepare it for model input:
    - **Remove Punctuation**: All non-word characters (such as punctuation marks) are removed from the text to standardize it.
    - **Whitespace Normalization**: Multiple spaces are replaced with a single space.
    - **Convert to Lowercase**: All text is converted to lowercase to make the model case insensitive.
    - **Text Deduplication**: In some cases, repeated words were removed to avoid redundancy and focus on unique content.
- **For CLIP method**:
    - **Truncation and Token Limitation**: For longer texts, the length is capped (e.g., 77 tokens for the CLIP-based method) to ensure the input fits within model limitations.

Effect of Pre-processing on Performance

Pre-processing the text inputs significantly impacts the performance of VLM-based image retrieval. Using photo descriptions alone achieves a Recall@30 of 0.693, indicating that these descriptions are well-aligned with dialogue content. In contrast, VLM captions alone result in a lower Recall@30 of 0.526, likely due to their brevity and lack of detail. VLM object lists improve performance to 0.612 by providing more structured and specific information about image contents.

Combining multiple text inputs boosts performance. Adding photo descriptions and VLM captions together increases Recall@30 to 0.723, as captions complement the detailed descriptions. Including all three inputs—photo descriptions, VLM captions, and VLM object lists—yields the highest performance of 0.736, highlighting the value of diverse and detailed textual information for accurate image retrieval.

2. How did you align the photo and dialogue text in the same embedding space? Use pretrained model or train your own?

Method 1 (VLM-based):

The photo and dialogue text are aligned in the same space using the embeddings generated by a vision-language model (LLaVA in this case). Each image gets a caption describing it in natural language. The caption is then pre-processed and embedded into the same space as the dialogue text. The dialogue text is processed similarly, and both are compared based on cosine similarity to find the most relevant image.

Method 2 (CLIP-based):

CLIP already aligns both text and images into the same embedding space. When using CLIP, both the dialogue text and images are passed through the model to generate their embeddings. CLIP's architecture ensures that both modalities (image and text) are represented in a shared space, where the relationship between the two can be directly compared using cosine similarity.

The CLIP model used is a pre-trained model from OpenAI, which has already been trained on vast amounts of text-image pairs, making it highly capable of aligning text and images without needing additional fine-tuning.

3. How do you improve the performance of your model?

To enhance Method 1, replacing LLaVA with a more advanced vision-language model like BLIP or FLAVA could improve captioning accuracy and object detection, leading to better alignment with dialogue content. For Method 2, fine-tuning CLIP on a task-specific dataset could significantly boost performance, as the current pre-trained model offers general but not domain-optimized results.

Introducing a multimodal fusion module can further improve both methods by combining image and text embeddings using advanced techniques like attention mechanisms, rather than relying solely on cosine similarity. This approach would allow the models to learn more intricate relationships between images and dialogue text. Additionally, incorporating contextual embeddings by including the broader dialogue history could provide richer context, improving retrieval relevance.

Lastly, data augmentation could enhance robustness. Augmenting dialogues with paraphrasing and rewording improve text generalization, while applying transformations like cropping or flipping to images helps handle diverse scenarios. Incorporating user metadata, such as historical image-sharing patterns, could also personalize retrieval, making it more relevant to user-specific contexts.