COSE474-2024F: Final Project Report
Evaluating the Performance of CLIP in Handling Negative Sentences for Visual-Language Matching

2022320077 Suhyun Kim [1]

1. Introduction

1.1. Motivation

CLIP is trained by 400 million data set from the internet. However, because of the dataset, it has been impacted by social context. Bias is an unavoidable problem with VLM. Although there are many cases, my interest is related with positive bias. With similar context, when LLM analyzes the emotion data, the model had predicted the positive better than the negative emotion because of the train dataset(Zhao et al., 2023). Also, the (Zhang et al., 2024) said LVLMs had bias related to social-cultural context. In psychology, there is a Pollyanna Principle, that people are tilted to positive semantic. My question starts from here. Does the VLM, especially, CLIP can handle the negative expressions? As CLIP is trained by positive syntax, I'm wondering the negative syntax can be connected to negative semantic in CLIP. I want to understand the CLIP, and evaluate performance with positive and negative captions(or prompts).

1.2. Problem definition

Vision-language models like CLIP have become a cornerstone in multimodal learning, excelling in tasks such as zero-shot classification, retrieval, and image-text alignment. However, despite its widespread use and proven performance in various scenarios, its ability with processing negative sentences is not fully detected. Negative prompts, such as "This is not a cat, just a dog", differ fundamentally from positive prompts. Negative sentences are semantically more complex because of not just the semantic meaning but also logical negation is contained in context. Understanding this apply for the field that needs precise textual understanding, for example, medical imaging or in autonomous driving. With its evaluation about negative sentence, we can give a guide for the enhancing the CLIP's textual ability. This study aims to systematically evaluate CLIP's performance in processing negative prompts, identifying its strengths, limitations, and potential for improvement. By shedding light on this less-explored area, we provide the foundational analysis necessary for advancing CLIP's robustness and ensuring its applicability in real-world, high-stakes environments.

2. Methods

This study is, initially, to systematically evaluate CLIP's capacity to handle negation in textual prompt.
This study's contributions include:

- Proposing a method to evaluate and analyze the similarity scores of image-text pairs, focusing on the divergence between positive and negative prompts.
- Introducing a framework for improving the differentiation between positive and negative prompts through enhanced prompt engineering.

This study has two major challenges. First, observation showed that CLIP tends to generate similar embedding vectors for positive and negative textual prompts, leading to high similarity scores even for semantically opposite meanings. Second, CLIP's training dataset primarily focuses on positive associations, making it less robust in processing negation. Following is how this study address them. To systematically evaluate, I used simple negation("This is not a cat with white belly."). As it came up with small difference, using multiple negative syntax to compare the similarity between prompts help to reason intensively. Also, not just qualitative analysis of similarity, I add quantitative analysis of similarity(etc.visualization).

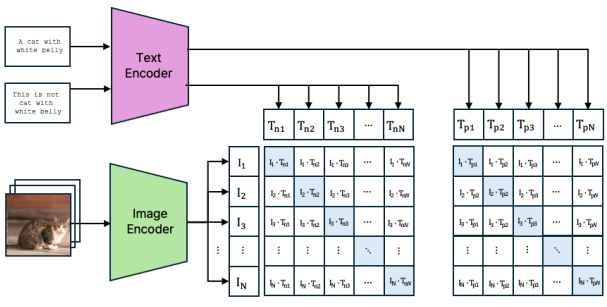

Figure 1. CLIP Architecture: Dual encoder structure for image and text inputs. Two inputs for text encoder. Each one is positive caption and negative caption in simple syntax difference.

$$S(I,T) = \frac{F_{\text{image}}(I) \cdot F_{\text{text}}(T)}{\|F_{\text{image}}(I)\| \cdot \|F_{\text{text}}(T)\|}$$

where: I: Input image. T: Input text prompt. $F_{\text{image}}(I)$: Feature vector of the image encoded by CLIP.$F_{\text{text}}(T)$: Feature vector of the text encoded by CLIP. $\|F_{\text{image}}(I)\|$: L_2-norm of the image feature vector. $\|F_{\text{text}}(T)\|$: L_2-norm of the text feature vector.

Algorithm 1 Similarity Evaluation with CLIP

Require: Image dataset $\mathcal{I}$, Text prompts $\mathcal{P}^+$ (positive), $\mathcal{P}^-$ (negative), CLIP model M
Ensure: Similarity scores S
for each image $i \in \mathcal{I}$ **do**
  $F_{\text{image}} \leftarrow M$.encode_image$(i)$
  Normalize $F_{\text{image}} \leftarrow F_{\text{image}}/\|F_{\text{image}}\|$
  for each prompt $p \in \mathcal{P}^+ \cup \mathcal{P}^-$ **do**
    $F_{\text{text}} \leftarrow M$.encode_text$(p)$
    Normalize $F_{\text{text}} \leftarrow F_{\text{text}}/\|F_{\text{text}}\|$
    $s \leftarrow F_{\text{image}} \cdot F_{\text{text}}$ ▷ Cosine similarity
    Add s to S
  end for
end for
return S

Algorithm 2 Similarity Comparison with Multiple Prompts

Image dataset $\mathcal{I}$, List of prompts $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, CLIP model M
Similarity scores S for each image-prompt pair
for each image $i \in \mathcal{I}$ **do**
  $F_{\text{image}} \leftarrow M$.encode_image$(i)$
  Normalize $F_{\text{image}} \leftarrow F_{\text{image}}/\|F_{\text{image}}\|$
  for each prompt $p \in \mathcal{P}$ **do**
    $F_{\text{text}} \leftarrow M$.encode_text$(p)$
    Normalize $F_{\text{text}} \leftarrow F_{\text{text}}/\|F_{\text{text}}\|$
    $s \leftarrow F_{\text{image}} \cdot F_{\text{text}}$ ▷ Cosine similarity
    Add (i, p, s) to S ▷ Store image, prompt, and similarity score
  end for
end for
return S

3. Experiments

3.1. Experimental design & setup

Main purpose of this experiment is to evaluate the performance of the CLIP(Radford et al., 2021) model in handling negative prompts(sentences) effectively. Using the MS-COCO(Lin et al., 2014) val2017 dataset, I compare the cosine similarity between image embeddings and textual prompts, which include positive descriptions and simple negation prompts in syntax level('This is not a original caption'). The experiment is conducted on a Google Colab environment with T4 GPU, pytorch v.2.5.1. Also, used the ViT-B/16,ViT-L/14@336px, ViT-B/32, RN50x16,ViT-L/14 backbone of CLIP. After the evaluating the performance, I made multiple prompts to be specific. These prompts are conducted as followings:

- {original caption}
- This is {original caption}
- Not {original caption}
- This is not {original caption}
- There is absolutely no {original caption}
- There is {change main subject in original caption}
- This is not {original caption};{extra explanation but doesn't match}

With these prompts, I also calculate the similarities and compared.

3.2. Result

Backbone	Positive Mean	Negative Mean	Positive Std	Negative Std
RN50x16	0.3067	0.3028	0.0370	0.0339
ViT-B/16	0.3079	0.2961	0.0341	0.0303
ViT-B/32	0.3041	0.2909	0.0334	0.0299
ViT-L/14	0.2571	0.2429	0.0390	0.0346
ViT-L/14@336px	0.2633	0.2506	0.0394	0.0347

Table 1. Comparison of Positive and Negative Means and Standard Deviations across Backbones.

Table 1 shows the similarity between image-text features. All the five backbone doesn't had remarkable difference in positive and negative prompts. Mostly, negative prompt-image similarity is slightly lower than positive prompt-image. It's true that negative is less similar with the image, but to state that CLIP handles the negative prompt, it should be much lower than the original (positive) statements. With low performance, I checked the case which positive prompt similarity is lower than negative. All the caption was 25014 but, 4789 captions had opposite results.

Figure 2 shows the t-SNE visualization of the image and text embeddings generated by the CLIP model. Blue points represent image embeddings, while orange and green points represent positive and negative text embeddings, respectively.

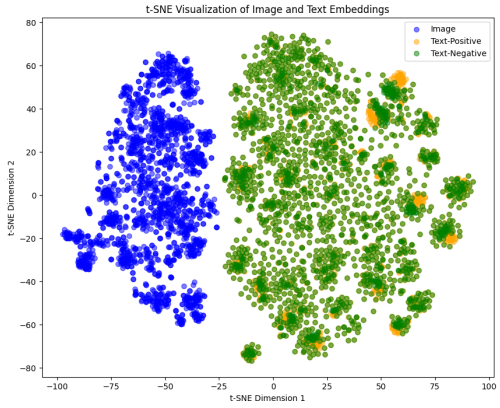

Figure 2. t-SNE Visualization of CLIP Embeddings: Images (blue), Positive Prompts (orange), Negative Prompts (green).

Negative prompts and positive prompts syntax is the same except the first three words. As the CLIP encoder works with token, negative text and positive text are mostly overlapped.

Figure 3 and Figure 4 shows the similarity between multiple prompt. To make the reasoning to be clear, I rather generate multiple prompts. For the image and caption selection, it is randomly processed in opposite cases, which positive similarity was lower. With simply adding 'not' gave more significant difference. This means CLIP can handle simple negation. However, with more grammatical syntax such as 'This is not', 'There is absolutely no' made similarity increase. Also, there is an interesting finding, which prompts with extra explanation but doesn't match, have the largest similarity.

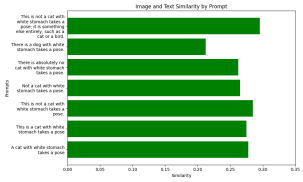

Figure 3. similarity for a cat image{ID:255965}.

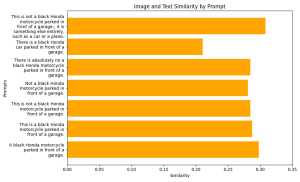

Figure 4. similarity for a motorcycle image{ID:179765}.

3.3. Discussion

Small difference in similarity score. The results of this study reveal interesting insights into CLIP's ability to process negative prompts. While the model shows a measurable distinction between positive and negative prompts, the overall difference in similarity scores is relatively small. This finding suggests that CLIP suffers to adequately differentiate between semantically positive and negative text inputs.
Similar embedding between positive and negative prompts. Across all backbones tested, the negative prompts produced similarity scores slightly lower than their positive counterparts. However, the magnitude of the difference was not substantial enough to confirm robust handling of negation by CLIP. The overlap in embedding spaces for positive and negative prompts, as shown in the t-SNE visualization, indicates that the tokenization mechanism in CLIP is not robust with semantic level. However, this is not surprising beyond the CLIP training mechanism. As text is tokenized by Byte-Pair encoding tokenizer, it will be depended on the structure, syntax-level.
Unexpected high similarity with contradictory prompts. Simple negation prompts such as "Not a cat with white stomach" resulted in more distinguishable embeddings compared to complex negations like "There is absolutely no cat with white stomach." This implies that CLIP's performance degrades with increasing grammatical complexity. Prompts with added contradictory explanations, such as "This is not a cat; it is a dog," surprisingly produced higher similarity scores. This could point to a bias in CLIP's training data, where phrases with explicit objects dominate the learned representations.

4. Future direction

Implications: These findings highlight the limitations of CLIP in accurately processing and embedding negation. This could pose significant challenges in high-stakes applications, such as medical imaging or autonomous systems, where precise textual understanding is critical. Moreover, the results emphasize the need for better token-level understanding and improved dataset representation of negation during training. Also, we can find that additional information with is mismatch can make the text-image similarity higher than original caption. This implicit that CLIP might have over-information bias either.

Future Directions: As, this experiment used 5000 images and approximately 25K simple negative captions to inference the CLIP's performance. More definitely with multiple prompt, we could make more negative caption set. Related to CLIP's performance, training dataset with more diverse and nuanced negative samples make the performance increase. We can not only augment the training data but also adjust the training weight of negative text tokens to significantly reduce their similarity, making them diverge further from simple positive syntax. Furthermore, investigating fine-tuning or prompt engineering techniques to enhance the semantic separation of positive and negative prompts will be one road we could walk through. On state-of-the-art, BLIP(Li et al., 2022) is also well-designed multimodal model, which itself generate the caption. Self-captioning is one kind of augmentation, means it might have better semantic level performance. Comparing the ability of CLIP and BLIP to handle negative sentences can also contribute

to understanding each model and their semantic processing mechanisms.

References

Fan, Z., Wei, Z., Li, Z., Wang, S., and Fan, J. Negative sample is negative in its own way: Tailoring negative sentences for image-text retrieval. *arXiv preprint arXiv:2111.03349*, 2021. doi: 10.48550/arXiv.2111.03349. URL https://doi.org/10.48550/arXiv.2111.03349.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. doi: 10.48550/arXiv.2201.12086. URL https://arxiv.org/abs/2201.12086.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Radford, A., Kim, J. W., Hallacy, A., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. doi: 10.48550/arXiv.2103.00020. URL https://doi.org/10.48550/arXiv.2103.00020.

Shtedritski, A., Rupprecht, C., and Vedaldi, A. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. doi: 10.48550/arXiv.2304.06712. URL https://arxiv.org/abs/2304.06712. Oral presentation.

Zhang, J., Wang, S., Cao, X., Yuan, Z., Shan, S., Chen, X., and Gao, W. Vlbiasbench: A comprehensive benchmark for evaluating bias in large vision-language models. *arXiv preprint arXiv:2406.14194*, 2024. doi: 10.48550/arXiv.2406.14194. URL https://doi.org/10.48550/arXiv.2406.14194.

Zhao, X., Liu, J., and Smith, N. A. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*, 2023. doi: 10.48550/arXiv.2305.15005. URL https://doi.org/10.48550/arXiv.2305.15005.