PMCoders at SemEval-2023 Task 1: RAltCLIP: Use Relative AltCLIP Features to Rank

Mohammad Javad Pirhadi, Motahhare Mirzaei,
Mohammad Reza Mohammadi and **Sauleh Eetemadi**
Iran University of Science and Technology at Tehran, Iran
{mohammad_pirhadi, m_mirzaei96}@comp.iust.ac.ir,
{mrmohammadi, sauleh}@iust.ac.ir

Abstract

Visual Word Sense Disambiguation (VWSD) task aims to find the most related image among 10 images to an ambiguous word in some limited textual context. In this work, we use AltCLIP features and a 3-layer standard transformer encoder to compare the cosine similarity between the given phrase and different images. Also, we improve our model's generalization by using a subset of LAION-5B. The best official baseline achieves 37.20% and 54.39% macro-averaged hit rate and MRR (Mean Reciprocal Rank) respectively. Our best configuration reaches 39.61% and 56.78% macro-averaged hit rate and MRR respectively. The code will be made publicly available on GitHub.

1 Introduction

WSD (Word Sense Disambiguation) is the task of identifying which sense of a word is meant in a sentence or other segment of text. In the VWSD (visual-WSD) task (Raganato et al., 2023), given a potentially ambiguous word and some limited textual context, the image corresponding to the intended meaning should be selected from a set of candidate images. Such a model can operate better as a text-to-image retrieval system.

CLIP (Radford et al., 2021) uses a contrastive loss function to map text and image features into a common space by minimizing the distance between the related caption and image and maximizing the distance between negative pairs. AltCLIP (Chen et al., 2022) alters the language encoder in CLIP and uses XL-R LM (Language Model) to extend its language capabilities. We use AltCLIP as the base model as the dataset includes Italian and Farsi languages in addition to English.

The official dataset consists of 12869, 16, 463, 305, and 200 samples as training, trial, English test, Italian test, and Farsi test data respectively. Each sample has a word (potentially ambiguous), a phrase including the word, and 10 images. To improve model generalization, we also use a subset of the LAION-5B (Schuhmann et al., 2022) dataset which consists of 5 billion image-caption pairs and has 3 parts: LAION-2B-en, LAION-2B-multi, and LAION-1B-nolang.

The experiments results show that the proposed model which relatively compares the images can be used to improve text-to-image retrieval systems by re-ranking the most related images.

2 System Description

Figure 1 shows an overview of the proposed model and the following subsections discuss it in more detail.

2.1 Feature Extraction

We map the phrase and images into a common space using AltCLIP (the AltCLIP model is frozen). In this stage, we have 11 vectors, 1 for the phrase and 10, each corresponding to an image.

2.2 Word Sense Disambiguation

"You shall know a word by the company it keeps" - J. R. Firth (1957)

Getting inspiration from this famous quote, we provide the model an extra feature vector by omitting the ambiguous word from the phrase and getting its feature vector using the AltCLIP text encoder and feeding it as the 12th feature vector (we call it WOAW, i.e. the phrase without the ambiguous word). For example, given the phrase "Andromeda tree" with "Andromeda" as the ambiguous word, by omitting the word "Andromeda" from the phrase "Andromeda tree" resulting in "tree", it's clear that the related image should be a "tree", not a "galaxy".

2.3 Relative Features

On top of AltCLIP, we use a 3-layer standard transformer encoder (Vaswani et al., 2017) to transform

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1751–1755

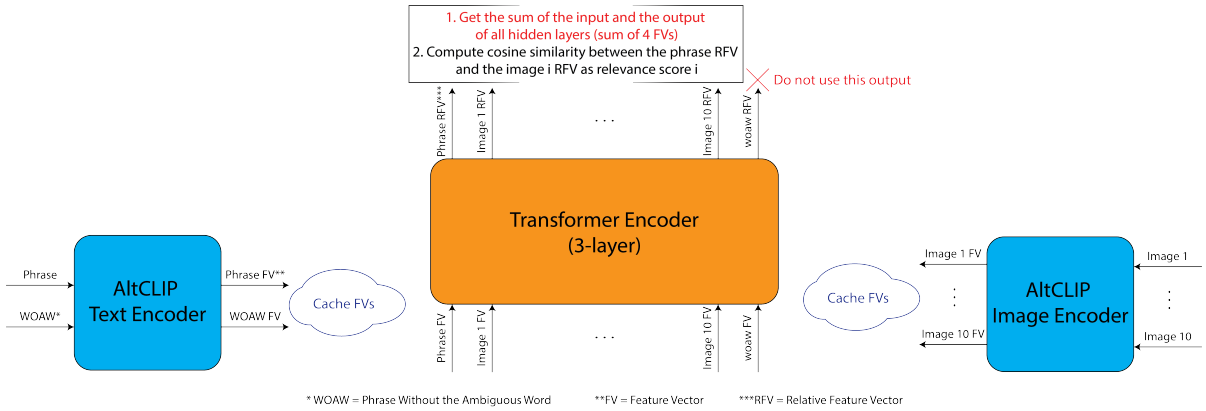

Figure 1: An overview of the proposed system. After calculating feature vectors (FVs) using AltCLIP (frozen) and caching them, we use a 3-layer transformer encoder to get relative feature vectors (RFVs). The relevance score for each image is calculated using cosine similarity between that image RFV and the phrase RFV.

the independent AltCLIP features for the text and images to relative ones. We clarify the intuition behind this using an example. Let's say there are 10 images with the same object. The objects are the same except in one property (e.g. 10 identical pens with different colors) and a phrase like "red pen". If the feature vector of the phrase is directly compared to the image feature vector, they would all be close. But if they are compared relatively using the phrase, the difference in color would become apparent, and the model would know to which property it should pay more attention.

2.4 Output

We sum the input and the output of all hidden layers (sum of 4 feature vectors) and then compute the cosine similarity between the phrase and each image using the calculated relative feature vector. The output is the most similar image to the text.

2.5 Loss Function

We explored the effect of using two loss functions: cross-entropy and cosine embedding loss with a margin of 0. See the section 4 for more information.

3 Experimental Setup

3.1 Pretraining

To improve the model generalizing, we used a subset of LAION-5B to pretrain the model. After filtering the dataset by caption length (<=50), caption word count (2 or 3) and image width (>=224), and height (>=224), we get 45,000 image-caption pairs for each language, extract their features using AltCLIP and use those features to calculate the similarities between all 45,000 images in each language. Then we choose the 9 most similar images to each image and form a dataset for pretraining that is like the official dataset but it's a little bit more challenging. For the 12th feature vector, we randomly drop one of the caption words and calculate its features.

3.2 Training Data Translation

The train and trial parts of the official dataset only include the English language but the test part includes English, Farsi, and Italian. To have Farsi and Italian samples at training time, we use Google Translate to translate the training and the trial part of the official dataset into Farsi and Italian.

3.3 Hyper-parameters

Table 1 shows the hyper-parameters at the pretraining and training stages. At the pretraining stage, we train the model for 30 epochs, and at fine-tuning stage, we train the model for 3 epochs. Note that as we use HuggingFace transformers, all of the unmentioned hyper-parameters have the default value.

3.4 Training Time Reduction

As we freeze the AltCLIP model, in all experiments and stages, we cache the calculated AltCLIP features once to reduce training time.

3.5 Framework & Tools

We use PyTorch (Paszke et al., 2019) + HuggingFace transformers (Wolf et al., 2020) to implement our models.

4 Results

To investigate the effect of the modifications, we examine the effect of using each one. You can see

Name	Value
attention dropout	0.5
dropout	0.5
weight decay	0.2
hidden activation	quick gelu
hidden size	768
intermediate size	3072
logit scale init value	2.6592
num attention heads	8
num hidden layers	3
batch size	256
init learning rate	5e-05

Table 1: Hyper-parameters used in pretraining and training stages

the results in table 2.

4.1 Pretraining

Except for pretraining using cosine embedding loss, pretraining increases both the average hit rate and average MRR (see next section for more details).

Pretraining reduces the English hit rate and MRR a little because the AltCLIP is pretrained on English and Chinese data so the extracted features for the English language using AltCLIP are good enough and pretraining the transformer encoder layer, reduces its focus on English data. We believe that pretraining with more data solve this problem.

For Italian and Farsi, pretraining increases both hit rate and MRR in almost all cases especially in cases where we do not use translated data.

4.2 Loss Function

We explored using cross-entropy (C) and cosine embedding loss (S) with a margin of 0 as loss functions. Cross-entropy can be considered as an extreme case of cosine embedding loss which tries to minimize the similarity as much as possible between negative samples and anchor (the phrase) while cosine embedding loss tries to push them away as much as the margin.

In the pretraining stage, because we select the 9 most similar images to the positive sample, the cosine embedding loss is not able to learn the data well (the training accuracy is about 40%); so when we fine-tune this model, the accuracy drops too much. **With this reason in our mind, we do not take the pretrained models using cosine embedding loss into account for our next analyses.**

For the rest of the setups, they do not differ too much but when we use translated data, cosine embedding loss is better and when we do not, cross-entropy loss is better. The reason is when we use translated data which is noisier, the generalization becomes more important and as a result, cosine embedding loss which can generalize better acts better.

4.3 The 12th Feature Vector

The 12th vector, in most cases, increase the average hit rate and MRR but sometimes using it decreases the accuracy a little. In fact, it causes the model to over-fit. Figure 2 shows an example of positive impact and an example of negative impact (over-fitting to the 12th vector). This problem can be solved by penalizing the usage of the 12th feature vector. This way the model only uses it when it really helps.

4.4 Translated Data

Without pretraining, for Farsi and Italian languages using translated data increase both hit rate and MRR in all cases but decreases them for English data (as mentioned in the pretraining subsection) because of the noise that exists in translated data. This reduction happens for Farsi and Italian languages with the pretraining stage exactly for the same reason; in other words, the translated data noise ruins the information that the model has gained during the pretraining stage. So using translated data only helps when the model does not have knowledge about some languages, at least in our case.

5 Future Work

There are some modifications that can be made to improve the overall accuracy of the proposed system, here are some of them:

- This system can be used as an image-to-text retrieval system after being pretrained using a massive amount of data.
- The 9 other images in pretraining samples can be selected by similarity to the caption feature vector instead of a related image (to caption) feature vector. It may be more challenging and effective.
- Instead of randomly dropping a word for the pretraining stage, a better word can be selected to drop.

Model	English		Italian		Farsi		Average	
	Hit Rate	MRR	Hit Rate	MRR	Hit Rate	MRR	Hit Rate	MRR
Random Baseline	12.53	31.67	11.50	31.27	6.23	26.07	10.09	29.67
Official Baseline	60.48	73.88	28.50	46.70	22.62	42.61	37.20	54.39
AltCLIP Zero-shot	64.58	77.08	27.87	47.39	8.00	31.55	33.48	52.01
P-C-11 Zero-shot	46.00	64.86	24.92	44.72	12.50	33.10	27.81	47.56
P-S-11 Zero-shot	7.78	27.44	9.18	28.86	7.00	25.22	7.99	27.18
P-C-12 Zero-shot	50.54	68.47	26.89	46.77	11.50	32.89	29.64	49.38
P-S-12 Zero-shot	7.99	27.10	11.15	29.48	6.00	24.77	8.38	27.11
C-11	**67.39**	**79.21**	30.16	48.95	12.50	34.86	36.68	54.34
S-11	64.79	77.48	32.13	51.06	11.00	33.56	35.98	54.04
C-11-ML	62.85	76.22	30.49	50.48	22.00	42.23	38.45	56.31
S-11-ML	64.58	77.49	30.49	50.90	21.00	41.46	38.69	56.62
C-12	67.17	79.14	30.16	48.96	12.50	34.92	36.61	54.34
S-12	64.58	77.39	31.80	51.15	12.00	34.12	36.13	54.22
C-12-ML	63.50	76.28	28.85	49.80	21.00	41.93	37.78	56.00
S-12-ML	64.15	77.30	30.49	50.96	19.50	40.24	38.05	56.17
P-C-11	65.44	78.29	31.48	50.55	17.50	38.89	38.14	55.91
P-S-11	33.69	52.63	19.34	39.59	12.00	31.11	21.68	41.11
P-C-11-ML	61.12	75.10	29.84	50.25	19.50	41.22	36.82	55.52
P-S-11-ML	46.44	63.60	21.31	42.75	13.00	31.14	26.92	46.50
P-C-12	65.87	78.34	**32.46**	**51.45**	20.50	40.54	**39.61**	**56.78**
P-S-12	27.21	47.65	19.67	38.42	12.50	30.77	19.80	38.95
P-C-12-ML	61.77	75.55	30.16	50.04	**22.50**	**43.49**	38.14	56.36
P-S-12-ML	35.42	55.40	22.30	42.02	12.00	31.40	23.24	42.94

Table 2: Results of different experiments. The best scores for each language are in bold.
(P = Pretrained, C = Cross-Entropy Loss, S = Cosine Embedding Loss, 11 = WOAW Not Used, 12 = WOAW Used, ML = Translated Data Used)

(a) Positive effect

(b) Negative effect

Figure 2: Figure a (left) shows an example of why the 12th feature vector can help. Given the phrase "Andromeda tree" with "Andromeda" as the ambiguous word, by omitting the word "Andromeda" from the phrase "Andromeda tree" resulting in "tree", it's obviously clear that the related image should be a tree, not a galaxy. And figure b (right) shows an example of why it can result in over-fitting. Given the phrase "Jungle lion" with "lion" as the ambiguous word, by omitting the word "lion" from the phrase "Jungle lion" resulting in "Jungle", we can see that the model predicted the jungle pictures as the most relevant pictures instead of lion picture which shows us over-fitting (The phrase and the word are translated from Farsi. For the original version, refer to 196th example of Farsi test data).

6 Conclusion

In this work, we proposed a system to relatively compare the similarity between a phrase and 10 images. Also, we used an extra feature vector to disambiguate the ambiguous word, Further more, we used a subset of LAION-5B to pretrain the proposed model to improve the generalization. The proposed system beat the official baseline by 2.41% and 2.39% in macro-averaged hit rate and MRR respectively. The results show that using a massive amount of data for pretraining, the proposed model can be used to improve text-to-image retrieval systems by re-ranking the most related images.

Acknowledgements

We would like to express our special thanks of gratitude to Dr. Behrouz Minaei who let us use his laboratory resources.

References

Zhongzhi Chen, Guang Liu, Bo-Wen Zhang, Fulong Ye, Qinghong Yang, and Ledell Wu. 2022. Altclip: Altering the language encoder in clip for extended language capabilities.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Alessandro Raganato, Iacer Calixto, Asahi Ushio, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2023. SemEval-2023 Task 1: Visual Word Sense Disambiguation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada. Association for Computational Linguistics.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.