Jina CLIP: Your CLIP Model Is Also Your Text Retriever

Andreas Koukounas Georgios Mastrapas Michael Günther Bo Wang Scott Martens Isabelle Mohr
Saba Sturua Mohammad Kalim Akram Joan Fontanals Martínez Saahil Ognawala Susana Guzman
Maximilian Werk Nan Wang Han Xiao

Jina AI
Leipziger str. 96, 10117 Berlin, Germany

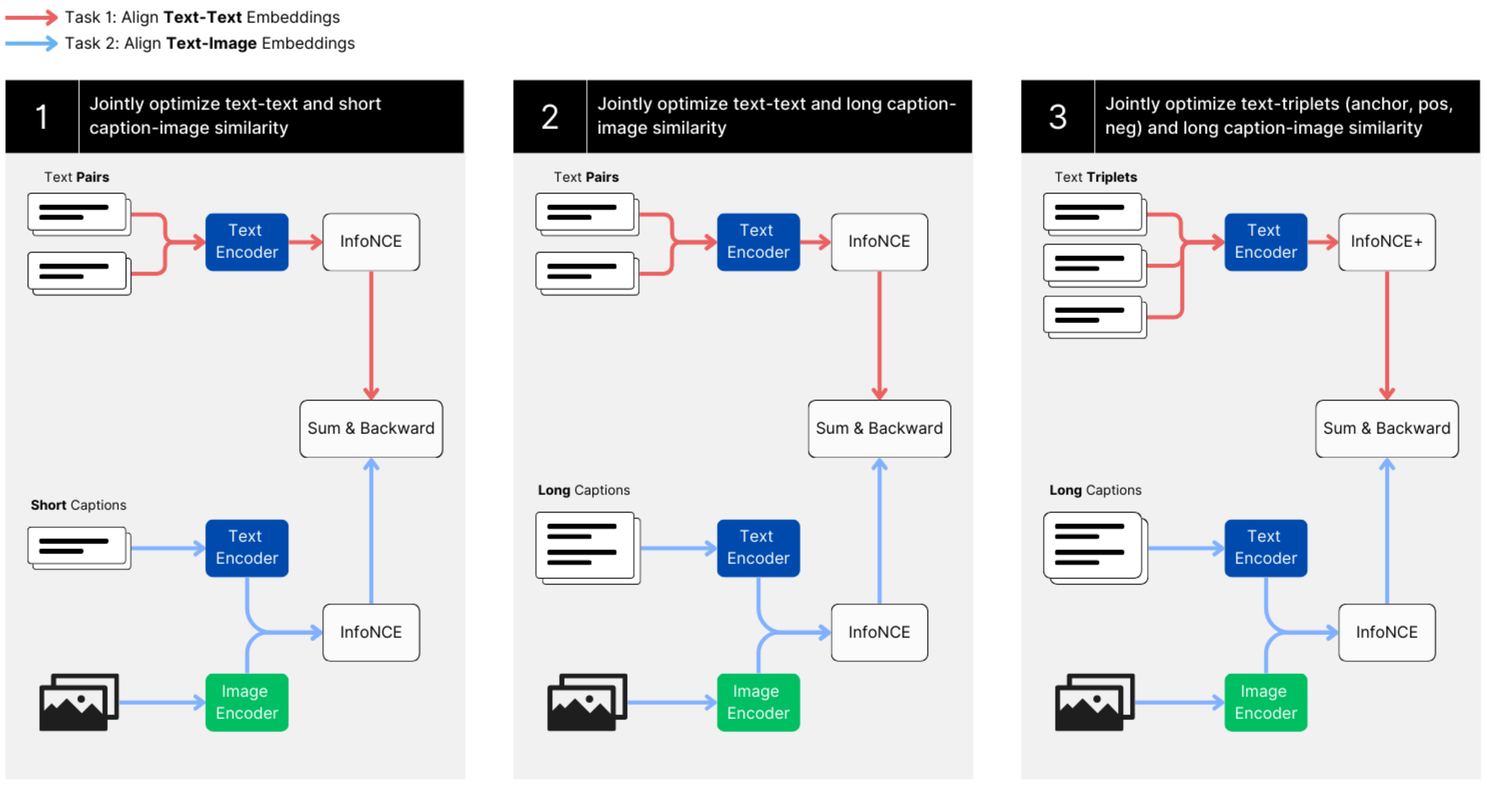

Figure 1: The training paradigm of `jina-clip-v1`, jointly optimizing text-image and text-text matching.

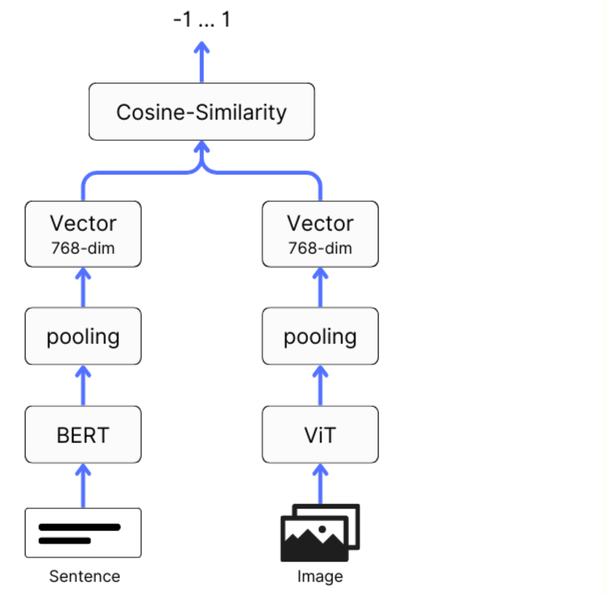

Figure 2: The model architecture of `jina-clip-v1`.

Introduction

Text-image contrastively trained models, such as CLIP, create an aligned representation space for images and texts by leveraging pairs of images and their corresponding captions. Similarly, text-text contrastively trained models, like `jina-embeddings-v2`, construct a representation space for semantically similar texts using pairs of related texts such as question/answer pairs, query/document pairs, or other text pairs with known semantic relationships.

In this paper, we present and demonstrate the effectiveness of a novel approach to contrastive training with large-scale image-caption pairs and text pairs. We jointly optimize for representation alignment of both text-image and text-text pairs, enabling the model to perform well at both kinds of tasks.

The resulting model, `jina-clip-v1`, performs comparably to CLIP-like models on the cross-modal CLIP Benchmark, while the text encoder by itself performs as well as similar models on MTEB Benchmark tasks.

Model Architecture

We opt for the dual encoder architecture introduced in the original CLIP. It comprises a text encoder and an image encoder that generate representations of identical dimensionality.

The text encoder uses the JinaBERT architecture. Experimental results indicate that starting from a model pretrained on Masked Language Modeling, yields superior final performance compared to starting from a text embedding model that has already been fully trained using contrastive learning.

For the image encoder, we use the EVA02 architecture. To keep the model size comparable to the text encoder, we select the base variant and initialize our model with the EVA02 pre-trained weights.

Training

Figure 1 illustrates our multi-task, three-stage training approach. This method jointly optimizes the model to perform two tasks: text-image matching and text-text matching.

- **Stage 1** focuses on learning to align image and text representations while minimizing losses in text-text performance. To this end, we train on large-scale and weakly supervised text-image and text-text pair datasets.
- **Stage 2** presents longer, synthetic image captions to the model while continuing to train with text-text pairs.
- **Stage 3** uses hard negatives to further improve the text encoder in separating relevant from irrelevant text. To maintain text-image alignment, we continue training on long image captions.

Data Preparation

Our text pair corpus consists of data from a diverse collection of 40 text-pair datasets and is used for text-text contrastive training during stages 1 and 2.

In stage 3 we use a triplet text corpus that includes hard negatives. This corpus combines data from MSMarco, Natural Questions (NQ), HotpotQA and the Natural Language Inference (NLI) dataset. Each training batch contains one annotated positive and seven negative items. We select hard negatives using text retrieval models to emphasize relevance in text triplets, except for NLI where negatives are chosen randomly.

For our text-image pairs in Stage 1, we use the LAION-400M dataset. LAION-400M contains 400M image-text pairs derived from Common Crawl.

In Stages 2 and 3, we use the ShareGPT4V dataset as our multimodal corpus. This dataset contains approximately 1.2M long captions generated by LLMs.

Loss Functions

The contrastive training on pairs, either text-image pairs or text-only pairs, optimizes the InfoNCE objective, as described in Equation 1. All stages and tasks, apart from the stage 3 text-text task, use this loss function.

$$\mathcal{L}_{\text{nce}}(\mathbf{B}) := \mathcal{L}_{\text{nce}}^{\rightarrow}(\mathbf{B}) + \mathcal{L}_{\text{nce}}^{\leftarrow}(\mathbf{B}), \text{ with}$$
$$\mathcal{L}_{\text{nce}}^{\rightarrow}(\mathbf{B}) := \mathbb{E}_{(\mathbf{q},\mathbf{p})\sim\mathbf{B}}\left[-\ln \frac{e^{\cos(\mathbf{q},\mathbf{p})/\tau}}{\sum_{i=1}^{k} e^{\cos(\mathbf{q},\mathbf{p_i})/\tau}}\right]$$
$$\mathcal{L}_{\text{nce}}^{\leftarrow}(\mathbf{B}) := \mathbb{E}_{(\mathbf{q},\mathbf{p})\sim\mathbf{B}}\left[-\ln \frac{e^{\cos(\mathbf{p},\mathbf{q})/\tau}}{\sum_{i=1}^{k} e^{\cos(\mathbf{p},\mathbf{q_i})/\tau}}\right] \quad (1)$$

Triplet training on texts with hard negatives optimizes the extended InfoNCE objective, given in Equation 2.

$$\mathcal{L}_{\text{nce}^+}(\mathbf{B}) := \mathbb{E}_{r\sim\mathbf{B}}\left[-\ln \frac{e^{\cos(\mathbf{q},\mathbf{p})/\tau}}{\sum_{i=1}^{k}\left[e^{\cos(\mathbf{q},\mathbf{p_i})/\tau} + \sum_{j=1}^{7} e^{\cos(\mathbf{q},\mathbf{n_{j,i}})/\tau}\right]}\right] + \mathbb{E}_{r\sim\mathbf{B}}\left[-\ln \frac{e^{\cos(\mathbf{p},\mathbf{q})/\tau}}{\sum_{i=1}^{k} e^{\cos(\mathbf{p},\mathbf{q_i})/\tau}}\right]$$
$$\text{with } r = (\mathbf{q}, \mathbf{p}, \mathbf{n_1}, \ldots, \mathbf{n_7}). \quad (2)$$

Training Steps

Each stage sums the losses of two tasks: one text-image alignment task, and one text-text alignment task:

$$\mathcal{L}_1(\mathbf{B}_{text;s}, \mathbf{B}_{img;s}) := \mathcal{L}_{\text{nce}}(\mathbf{B}_{text;s}) + \mathcal{L}_{\text{nce}}(\mathbf{B}_{img;s})$$
$$\mathcal{L}_2(\mathbf{B}_{text;l}, \mathbf{B}_{img;l}) := \mathcal{L}_{\text{nce}}(\mathbf{B}_{text;l}) + \mathcal{L}_{\text{nce}}(\mathbf{B}_{img;l})$$
$$\mathcal{L}_3(\mathbf{B}_{text3}, \mathbf{B}_{img;l}) := \mathcal{L}_{\text{nce}}(\mathbf{B}_{text3}) + \mathcal{L}_{\text{nce}^+}(\mathbf{B}_{img;l}) \quad (3)$$

Evaluation

Benchmark	**CLIP Benchmark**		**MTEB**		
Task Type	Zero-Shot Retrieval		Retrieval	STS	Average
Model - Metric	txt-img r@5	img-txt r@5	ndcg@10	spearman	ccore
OpenAI CLIP ViT B/16	75.62	88.12	17.63	66.22	43.95
EVA-CLIP ViT B/16	**82.15**	90.59	26.03	69.62	47.64
LongCLIP ViT B/16	81.72	**90.79**	28.76	68.57	47.71
`jina-embeddings-v2`	-	-	47.85	80.70	**60.38**
`jina-clip-v1` stage 1	78.05	86.95	39.52	77.96	56.51
`jina-clip-v1` stage 2	81.86	90.59	40.44	78.33	57.19
`jina-clip-v1`	80.31	89.91	**48.33**	**80.92**	60.12

Table 1 shows our evaluation results. To evaluate the model's cross-modal performance, we use the CLIP Benchmark which includes zero-shot image-classification and zero-shot cross-modal retrieval tasks. `jina-clip-v1` achieves an average Recall@5 of 85.8% across all retrieval benchmarks, outperforming OpenAI's CLIP model and performing on par with EVA-CLIP.

To evaluate `jina-clip-v1`'s text encoder, we use the Massive Text Embedding Benchmark (MTEB). CLIP-like models generally perform poorly on text embedding tasks, particularly information retrieval. However, `jina-clip-v1` competes closely with top-tier text-only embedding models, achieving an average score of 60.12%.

Further Information

Jina CLIP:
Koukounas, A., Mastrapas, G., Günther, M., Wang, B., Martens, S., Mohr, I., Sturua, S., Akram, M.K., Martínez, J.F., Ognawala, S., Guzman, S., Werk, M., Wang, N., & Xiao, H. (2024). Jina CLIP: Your CLIP Model Is Also Your Text Retriever. *ArXiv, abs/2405.20204*.

Embedding API:
`https://jina.ai/embeddings/`

Get the `jina-clip-v1` model on HuggingFace
`https://huggingface.co/jinaai/jina-clip-v1`

Workshop on Multi-modal Foundation Model meets Embodied AI (MFM-EAI) 26 Jul 2024 @ ICML 2024 in Vienna