Stitch

AML Challenge Report 2025/26

Davide Perniconi
Matricola: 1889270

Daniele Marretta
Matricola: 1985747

Olga Corovencova
Matricola: 2249558

Leonardo Lavezzari
Matricola: 1984079

1 Proposed Method

To achieve the ultimate goal outlined in the challenge, the team explored several approaches to find the best solution. Ultimately, only one model proved more successful than the others, resulting in a **MoCo-inspired Space Translator**[1] approach. We trained with a *contrastive objective* over a large dynamic memory bank of negative samples. Our goal was to learn a mapping function $T : R^{1024} \implies R^{1056}$ able to correctly align the space of image embeddings with the captions of text embeddings.

- **Architecture:** Our adapter is based on an MLP with two hidden layers, LayerNorm and GELU activations, designed to provide a smooth non-linear transformation between embedding spaces:

  1. *Input dimension:* 1024
  2. *Hidden layers:* $[1024, 1056]$
  3. *output dimension:* 1536
  4. *Activation:* Gelu
  5. *Dropout:* 0.5
  6. *Inizialization:* Xavier uniform
  7. *Normalization:* the final embedding is L2-normalized

  ...allowing the model to dynamically have a better control of the InfoNCE distribution. The model achieves a balance between non-linearity and regularization, avoiding overfitting despite the high embedding dimension.

- **Loss Function:** We train the translator using a **MoCo-style InfoNCE contrastive loss**[1]. Given a minibatch of queries (translated text embeddings) and their corresponding positive keys (image embeddings), the loss is defined by:

  1. *Positive pair:* $l_{pos} = k_{pos} \cdot q$
  2. *Negatives:* obtained from a queue with $10,000$ images embedding
  3. *Similarity distribution:* $logits = [l_{pos}, l_{neg}] \cdot exp(logitScale)$
  4. *Final objective:* $\varphi = -log \cdot \frac{e^{<q,k_{pos}>/\tau}}{\sum_{i=1}^{\infty} e^{<q,k_i>/\tau}}$

  This type of loss allows the two types of embeddings to be as close as possible to each other, while the other negatives are far away.

- **Training Details:** We used the following training setup:

  Optimizer: AdamW

  Learning rate: 0.01

  Weight decay: $1e - 5$

  Batch size: 256

  Epochs: 250

  Scheduler: ReduceLROnPlateau (monitoring validation loss)

  Patience: 5 epochs

  Memory bank (queue): 10,000 negatives

  Early stopping: triggered on stagnation for validation loss

  Validation metrics: MRR and Recall@K

  At each iteration:

  - The model outputs the query embedding.
  - Keys are taken directly as ground-truth image embeddings.
  - The positive key is enqueued, and the oldest negative samples are dequeued.
  - Contrastive loss is computed.
  - Cosine similarity-based retrieval metrics are evaluated during validation.

2 Results and Discussion

On the validation set, the model obtained:

MRR: 0.XX

Recall@1: 0.XX

Recall@5: 0.XX

Recall@10: 0.XX

NDCG@100: 0.XX

These results outperform the baseline provided by the challenge, which reports an MRR of approximately 0.462. The improvement highlights the effectiveness of combining a non-linear adapter with a contrastive learning framework optimized specifically for retrieval. Several design choices contributed significantly to the model's strong performance :

1. **Contrastive InfoNCE loss directly optimizes retrieval quality**[1]
   Unlike regression-based baselines (MSE, cosine embedding loss), in our case the InfoNCE objective maximizes the similarity between a caption and its corresponding image while explicitly pushing away thousands of negatives. This helped us to get the optimization aligned with the evaluation metric.

2. **Large dynamic memory bank of negatives**
   Using a memory queue of 10,000 negative samples increases diversity and stabilizes contrastive learning.

3. **Non-linear MLP with LayerNorm and GELU**
   This enables the model to learn a flexible non-linear mapping between embedding spaces while maintaining stability and preventing overfitting.

4. **Retrieval-based validation loop**
   Although the training objective minimizes InfoNCE loss, model selection is performed using MRR and Recall@K, ensuring that the final checkpoint is chosen based on real retrieval performance.

3 Conclusion

Our MoCo-based translator in local testing improved retrieval performance by combining a non-linear mapping with contrastive learning over a large memory bank. This approach proved to be the most effective among our experiments, offering a robust and highly discriminative alignment between text and image embedding spaces.
LINK TO THE REPO

4 What We Tried

During the competition we found models that did not outperform our final MoCo-based method, instead they provided valuable insights and contributed to our overall understanding of the problem.

Method 1: Simple Linear Mapping

A single linear projection trained with MSE loss. This baseline performed poorly at the beginning, confirming that the mapping between text embeddings and image embeddings is highly non-linear.

Method 2: Deep 5-Layer MLP

A deeper MLP architecture with 5 hidden layers and ReLU activation. Despite its higher capacity, this model overfitted the training set and achieved lower validation performance than simpler architectures, suggesting that depth without strong regularization is insufficient for this task.

Method 3: Two-Head MLP with Memory Bank (Second best model)

We explored a more sophisticated approach inspired by vector decomposition. The model consists of two parallel heads, trying in the unnormalization of the output:

- a **Direction Head** predicting a unit-norm embedding
- a **Scale Head** predicting a positive scalar via Softplus

The output embedding is computed as

$$T(x) = \text{normalize}(d(x)) \cdot s(x).$$

This factorized representation allows the model to decouple angular and radial components (direction/lenght), potentially capturing finer structure in the embedding space. Training was performed using a MoCo-style InfoNCE loss with a large negative queue (up to 32,000 samples). The model achieved competitive retrieval performance (MRR $\approx 0.XX$), outperforming standard regression methods. However, it did not surpass the simpler MoCo-based Space Translator, likely because scale modulation introduces additional instability in contrastive optimization.

Method 4: Optuna Hyperparameter Search

We conducted extensive hyperparameter optimization with Optuna, exploring activation functions, initialization schemes, dropout levels, queue sizes, and architecture depths. While Optuna improved stability and provided several useful configurations, the gains were incremental compared to the impact of the contrastive framework itself.

References

[1] Fabio Galasso. *Self-Supervised Learning. Advanced Machine Learning Course Slides, Sapienza University of Rome*. Lecture notes, 08_AML_SSL.pdf. 2025.