A Novel Attention-based Aggregation Function to Combine Vision and Language

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi and Rita Cucchiara

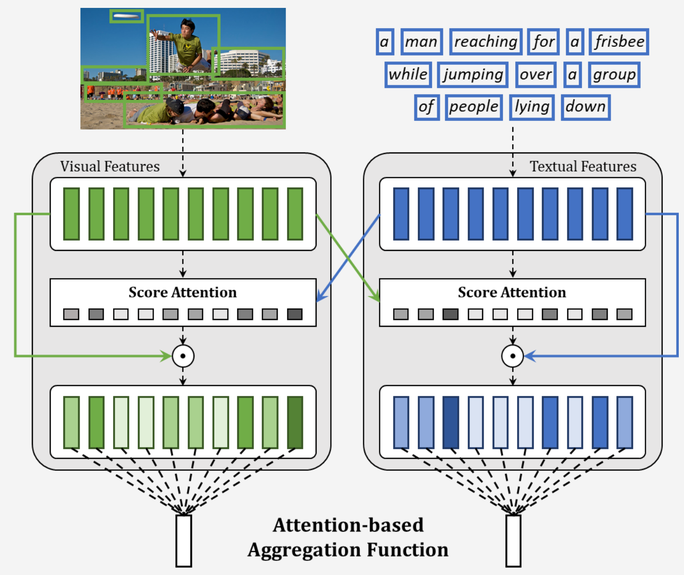
University of Modena and Reggio Emilia, Italy - name.surname@unimore.it

Overview

How can we effectively **aggregate** and **represent** sets or sequences with **Transformers** architectures [1]?
Usually this is done with the **CLS Token** [2][3], a special token appended at the beginning of each sequence.

To this aim we propose a new method with the following contributions:

- We introduce a **new aggregation function** based on **attention mechanisms** that learns a compact representation of sets or sequences of feature vectors.
- We tailor our method to combine vision-and-language data in order to obtain a **cross-modal reduction** for both classification and ranking objectives. We show its superior performance in **aggregating feature vectors** in multi-modal settings, compared to other common reduction operators.
- Also, our method can be **easily adapted** to other tasks requiring an aggregation of elements with minimum changes in the architecture design.



Aggregation Method

- Given two modality ***X*** and ***Z***, we compute a compressed vector for ***X*** as the weighted sum of its vectors:

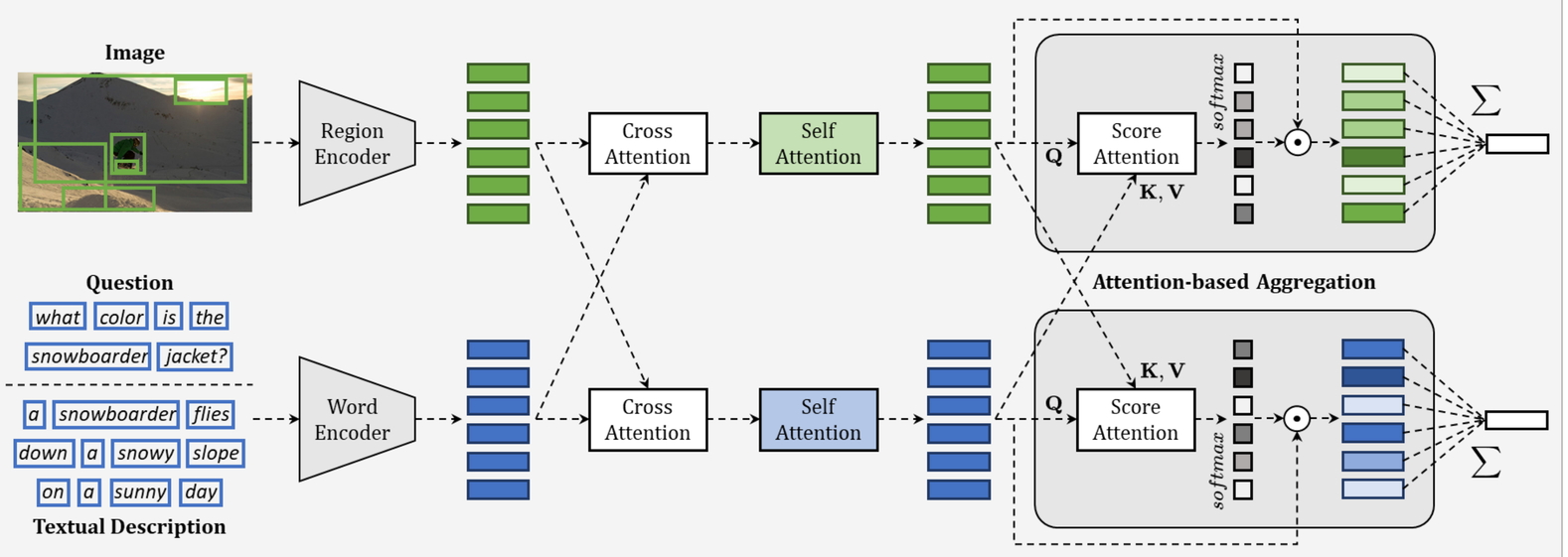
$$\boldsymbol{Y}(\boldsymbol{X},\boldsymbol{Z}) = \sum_{i=0}^{n_q} \boldsymbol{S}_i(\boldsymbol{X},\boldsymbol{Z}) \cdot \boldsymbol{X}_i$$

$$\boldsymbol{S}(\boldsymbol{X},\boldsymbol{Z}) = \mathrm{softmax}\left(\mathrm{ScoreAttn}(\mathcal{Q},\mathcal{K},\mathcal{V})\right)$$

- where each weight is a score computed with a function based on cross-attention mechanism, with ***Q*** projection of ***X*** and ***K***-***V*** projections of the other modality ***Z***:

$$\mathrm{ScoreAttn}(\mathcal{Q},\mathcal{K},\mathcal{V}) = \mathrm{fc}\left(\left[\mathrm{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d}}\right)\boldsymbol{V}_h\right]_h\right)$$

- We compare different aggregation functions on top of the same pipeline shown on the right.



VQA results

Aggregation Function	Validation All	Validation Yes/No	Validation Number	Validation Others	Test-Dev All	Test-Dev Yes/No	Test-Dev Number	Test-Dev Others
Mean Pooling	54.87	71.50	37.93	46.69	56.05	71.00	38.88	47.19
Max Pooling	56.73	75.68	37.64	47.37	57.95	75.14	38.48	47.69
LogSumExp Pooling	54.61	70.94	38.27	46.53	55.68	70.36	38.72	47.00
1D Convolution	56.87	72.35	39.18	49.79	57.79	71.71	39.97	49.96
CLS Token	58.31	74.29	39.89	51.03	59.40	74.26	40.31	51.07
Ours ($k=1$)	60.73	77.68	41.86	**52.84**	62.05	77.84	42.47	**53.03**
Ours ($k=2$)	60.76	78.06	42.32	52.48	62.06	78.26	42.62	52.66
Ours ($k=3$)	60.50	77.82	41.56	52.33	61.80	78.22	41.69	52.35
Ours ($k=5$)	**60.99**	**78.62**	42.53	52.46	62.17	78.52	42.27	52.74
Ours ($k=7$)	60.95	78.40	**42.65**	52.53	**62.43**	**78.75**	**43.33**	52.83
Ours ($k=10$)	59.94	77.30	40.82	51.80	61.16	77.39	40.69	51.97

Our aggregation method shows superior performances compared with the most common reduction operators.

Our function can be executed **multiple times** with different query, key and value projections, thus yielding **more** output vectors (*k*), that we finally average.

This can foster a more **disentangled** representation, in which different output vectors refer to different **global aspects** of the same input features.

For training we employ the **binary cross entropy** loss in a multi-label fashion, i.e. applying it independently for all classes.

Notably, we do not make use of any data augmentation strategy and do not employ any external data source.

Qualitative Results

Retrieval results

Aggregation Function	Text Retrieval R@1	Text Retrieval R@5	Text Retrieval R@10	Image Retrieval R@1	Image Retrieval R@5	Image Retrieval R@10
Mean Pooling	69.66	93.12	97.64	50.42	82.27	90.83
Max Pooling	69.04	92.68	96.98	51.20	83.27	91.52
LogSumExp Pooling	64.20	91.52	96.84	47.22	82.26	91.23
1D Convolution	65.66	91.86	96.58	49.25	81.43	90.42
CLS Token	70.30	93.38	97.24	51.05	83.28	**91.80**
Ours ($k=1$)	**70.80**	93.16	97.24	50.77	82.76	91.31
Ours ($k=2$)	70.36	**93.46**	97.20	**51.31**	**83.38**	91.69
Ours ($k=3$)	70.42	93.34	97.22	50.98	83.17	91.65
Ours ($k=4$)	70.14	93.42	**97.76**	50.82	82.66	91.14

Overall performance in cross-modal retrieval when generating different aggregated output vectors (*k*).

A lower number of vectors obtained the best results.

For each modality we produce ***k* compressed vectors** that we pair-wise compare with cosine similarity, and finally we average the resulting *k* similarity scores.

Intuitively, each aggregation module learns to extract and compare different relevant information, **specializing** each vector to distinct semantic meaning.

For training we adopt the contrastive **triplet ranking loss**, considering only the hardest negatives found in the mini-batch following [4].

Qualitative Results

References

[1] Vaswani, et al. “Attention is all you need” NeurIPS, 2017
[2] Devlin, et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” *ACL ,2019*
[3] Hao, et al. “LXMERT: Learning Cross-Modality Encoder Representations from Transformers” EMNLP, 2019.
[4] Faghri, et al. “VSE++: Improving Visual-Semantic Embeddings with Hard Negatives” BMVC, 2018.