Token-Type Embeddings (TTE)

In order for the shared Transformer architecture to work, the shared Transformer block needs to be able to somewhat differentiate between both modalities (image and text). Even though we desire an aligned representation to form in the shared block, especially the Self-Attention mechanism still needs to be able to differentiate between the two modalities. This can be explained by the fact that for images the model needs to find 2-dimensional spatial relationships, while for text only 1-dimensional relationships are required. Even though we saw in previous experiments that the shared Transformer block is able to learn a good representation without any modality-specific information, we still want to investigate how the performance of the model changes when we explicitly provide the shared Transformer block with the information which modality it is processing. This nessessitates a special embedding for both image and text, which is added to each token of the respective modality, even the special tokens. Our motivation in this change lies in the fact that VLMo [1] also follows this approach.

The intuitive approach to implement this, would be to directly follow VLM [1] and add the token type embeddings after the positional encoding, before the input is fed into the Transformer blocks [1]. Our definition of the Transformer input changes as follows for text:

$$\begin{aligned} \boldsymbol{H}_0^w &= \left[\boldsymbol{w}_0^{[\texttt{T\_CLS}]}, \boldsymbol{w}_0^1, ..., \boldsymbol{w}_0^M, \boldsymbol{w}_0^{[\texttt{T\_SEP}]}\right] + \boldsymbol{T}_{\text{pos}}^w + \boldsymbol{T}_{\text{type}}^w, \text{with} \\ \boldsymbol{T}_{\text{pos}}^w &= \left[\boldsymbol{t}_{\text{pos}_{[\texttt{T\_CLS}]}}^w, \boldsymbol{t}_{\text{pos}_1}^w, ..., \boldsymbol{t}_{\text{pos}_M}^w, \boldsymbol{t}_{\text{pos}_{[\texttt{T\_SEP}]}}^w\right], \text{and} \\ \boldsymbol{T}_{\text{type}}^w &= \left[\boldsymbol{t}_{\text{type}_{[\texttt{T\_CLS}]}}^w, \boldsymbol{t}_{\text{type}_1}^w, ..., \boldsymbol{t}_{\text{type}_M}^w, \boldsymbol{t}_{\text{type}_{[\texttt{T\_SEP}]}}^w\right] \end{aligned} \tag{1}$$

Similar holds for images:

$$\begin{aligned} \boldsymbol{H}_0^v &= \left[\boldsymbol{v}_0^{[\texttt{I\_CLS}]}, \boldsymbol{v}_0^1, ..., \boldsymbol{v}_0^N\right] + \boldsymbol{T}_{\text{pos}}^v + \boldsymbol{T}_{\text{type}}^v, \text{with} \\ \boldsymbol{T}_{\text{pos}}^v &= \left[0, \boldsymbol{t}_{\text{pos}_1}^v, ..., \boldsymbol{t}_{\text{pos}_N}^v\right], \text{and} \\ \boldsymbol{T}_{\text{type}}^v &= \left[\boldsymbol{t}_{\text{type}_{[\texttt{I\_CLS}]}}^v, \boldsymbol{t}_{\text{type}_1}^v, ..., \boldsymbol{t}_{\text{type}_N}^v\right] \end{aligned} \tag{2}$$

The token type embedding is the same for every token and patch in the sequence, respectively.

$$\begin{aligned} \boldsymbol{t}_{\text{type}_i}^w = \boldsymbol{t}_{\text{type}_j}^w, \forall i, j \in \{[\texttt{T\_CLS}], 1, ..., M, [\texttt{T\_SEP}]\} \\ \boldsymbol{t}_{\text{type}_i}^v = \boldsymbol{t}_{\text{type}_j}^v, \forall i, j \in \{[\texttt{I\_CLS}], 1, ..., N\} \end{aligned} \tag{3}$$

This follows VLMo [1] and is the case because each token/patch of a text/image input sequence is of the same modality. The parameters added to the model are negligible, as they only include two additional embeddings of size D, with $D = 768$ this accounts for just $768 * 2 = 1536$ parameters.

We present the results in Table 1, and show that the variant with TTE does not improve the performance of the model.

Model	ImageNet-1K Top-1 Accuracy	ImageNet-1K Top-5 Accuracy	Retrieval MSCOCO	Retrieval Flickr30K
EMKUM	26.1	50.4	66.3	77.83
$\text{EMKUM}_{\text{TTE}}$	25.6	49.5	66.0	77.36

Table 1: Introducing token-type embeddings (TTE) [1] after the positional encoding degrades performance slightly. We compare to the model variant introduced in (TODO: cite "Projections for Task-Seperation"), which does not use TTE.

We suspect that this is due to two reasons. First, the the TTE is added before the modality-specific encoders, so image and text encoder. Their task is to extract features from the input, independent of the other respective modality. So TTE is of no use to them, as the same embedding is added to every token/patch, and one does not need to differentiate between image and text in the modality-specific encoders. Second, even worse, we use pretrained models for the encoders (Data2Vec2), which already extract rich features from the input. Adding a TTE to their input, which is randomly initialized, will destroy the information the modality encoders receive as input, and thus the feature extraction will be less effective. Even though we are also training the pretrained image and text encoder, it will take time until the encoders learn to adapt to the TTE, and until the TTE has been trained in general.

We therefore opt for two changes to the TTE approach. First, we add the TTE after the modality-specific encoders, meaning the token type embedding is added to the output of the image and text encoder. This way, the TTE will not disturb the feature extraction of the image and text encoder. However, what will inevitably happen is that addind the TTE after the modality-specific encoders will destroy the features extracted by the aforementioned encoders, which is exactly the second problem mentioned before.

A possible solution can be found in the Transformer block of extremely deep and large models, such as BEiT-3 [2]. They multiply the output of the Self-Attention and Feed-Forward in a Transformer block with a learnable scalar parameter per embedding dimension. What makes this approach special is that the scalar parameters are all initialized with a values close to zero. As this multiplication is done before the residual connection, i.e. the addition of the input the the Self-Attention and Feed-Forward respectively, the contribution of the Self-Attention and Feed-Forward to the output of the Transformer block is very small at the beginning of training. What follows is that the initial input to the model is carried very far through the model, and the actual contribution of the parameters is added as the model lears to extract meaningful features from the input [3].

$$x'_l = x_l + \text{diag}(\lambda_{l,1}, \ldots, \lambda_{l,d}) \times \text{SA}(\eta(x_l))$$
$$x_{l+1} = x'_l + \text{diag}(\lambda'_{l,1}, \ldots, \lambda'_{l,d}) \times \text{FFN}(\eta(x'_l))$$

Figure 1: LayerScale adds a per-channel learnable weight to the output of Self-Attention and FFN. This can be modeled as a matrix with the diagonal being the learnable weights. With weights close to zero, the second term, i.e. the result of the feature extraction will have close to no contribution to the output of the Transformer block, leading the initial input to dominate during the initial stages of training. The input will therefore be carried very deep into the network [3].

We will use LayerScale not for the purpose it was intended for, that is to allow training of extremely deep models [3], but to allow the TTE to be added after the modality-specific encoders, without destroying the features extracted by the encoders. We initialize the weights of the LayerScale to 1e-5, meaning the contribution of the TTE to the extracted features will be almost zero at the beginning. The shared Transformer block will receive the almost unaltered features, and as the scale is learned, the TTE will be added as the models sees fit - as much as it helps the model to learn.

$$\begin{aligned} \boldsymbol{H'}^{w}_{L} &= \boldsymbol{H}^{w}_{L} + \text{diag}\big(\lambda_{L,1}, ..., \lambda_{L,D}\big) * \boldsymbol{T}^{w}_{\text{type}}, \text{or simply} \\ \boldsymbol{H'}^{w}_{L} &= \boldsymbol{H}^{w}_{L} + \text{LayerScale}\big(\boldsymbol{T}^{w}_{\text{type}}\big) \end{aligned} \tag{4}$$

$$\begin{aligned} \boldsymbol{H'}^{v}_{L} &= \boldsymbol{H}^{v}_{L} + \text{diag}\big(\lambda_{L,1}, ..., \lambda_{L,D}\big) * \boldsymbol{T}^{v}_{\text{type}}, \text{or simply} \\ \boldsymbol{H'}^{v}_{L} &= \boldsymbol{H}^{v}_{L} + \text{LayerScale}\big(\boldsymbol{T}^{v}_{\text{type}}\big) \end{aligned} \tag{5}$$

We use the same LayerScale for both image and text type embeddings, as the contribution of the type embeddings should be the same for both modalities. We do not want to bias the model towards one modality.

Model	ImageNet-1K Top-1 Accuracy	ImageNet-1K Top-5 Accuracy	Retrieval MSCOCO	Retrieval Flickr30K
EMKUM	26.1	50.4	66.3	77.83
$EMKUM_{TTE}$	25.6	49.5	66.0	77.36
$EMKUM_{TTE'}$	26.7	50.9	66.6	78.09

Table 2: Moving the TTE after the modality-specific encoders and using LayerScale with a low contribution weight of 1e-5, we denote this model variant as $EMKUM_{TTE'}$.

< Check average value of LayerScale weights -> should not be close to 0 > Mean: 0.0007224410073831677 (initial 0.00001) Standard Deviation: 0.016118625178933144

Bibliography

[1] H. Bao *et al.*, "VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts," in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=bydKs84JEyw

[2] W. Wang *et al.*, "Image as a Foreign Language: BEIT Pretraining for Vision and Vision-Language Tasks," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 19175–19186. doi: 10.1109/CVPR52729.2023.01838.

[3] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going Deeper With Image Transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 32–42.