Multimodal Knowledge Distillation

Differences to Unimodal KD

Aligned Representations

Dual encoder

Unimodal Student

- we start as simple as possible
- we just want to create aligned cross-modal representations, i.e. have the same representation for the same concept -> same representation for an image and its corresponding text (image-text pair in the dataset)
- multiple architectures are possible, we start with a CLIP-like architecture
- means one encoder per modalitiy, so one for image and one for text
- usually trained for scratch -> but expensive, and we want to utilize already pretrained(!) (self-supervised, not trained on labeled data) unimodal models
- we use BEiT-2 as the teacher model, which is an image model, and at the same time serves as our image encoder
- to now align representations, we just train the text encoder to regress the cls token of the image encoder
- hope is that the cls token of the teacher, which is regressed, has learned a representation that is abstract enough so that it can be applied independently of the modality, in this case images
  - although there has been no incentive for the teacher model to learn which is independent of the modality, as the model has only been pretrained on images -> if the representation (cls token) does still contain image specific information, the student model will not be able to learn a meaningful representation of the text, as image and text are inherently different
- in the first experiment we just train the text encoder to regress the cls token of the image encoder, nothing more
- the text model is smaller than the image model, it contains only 7 layers, while the image model contains 12 layers
- hope is that through Knowledge-Distillation we do not need a model as large as the teacher model, as we have seen that a smaller model can achive a performance quite similar to the larger teacher model through KD in the unimodal case
  - whether this translates to an multimodal setting can be derived from the results of this experiment
  - this could be seen in the retrieval application of our model -> if the performance on text-image retrieval is similar to the performance of the teacher model, then we can assume that the student model has learned a meaningful representation of the text(?)
- we still have the option to expand our student model to the same size as the teacher model, i.e. 12 layers
- still relatively cheap, as we only have to train the text encoder and, as we are doing in the first experiment with 7 layers, we can initialize the text encoder (the student) with the weights of the text D2V2 model, meaning we do not start from scratch

Adding Image-Text Contrast

- until now we did not actually used the same philosophy as in CLIP, which relies on, next to a seperate image and text encoder, a contrastive loss to align the representations of the image and text, so does not do KD and trains both text and image encoder from scratch
- as mentioned in the chapter about CLIP, the architecture features two linear projections, one for each modality/encoder

- goal is to project the image/text represenation in a shared multimodal embedding/latent space, on which the contrastive loss is computed
- if we also manage to do this successfully, the performance on image-text retrieval should increase by a margin

Stagewise Unimodal Distillation

Seperate Self-Attention

Baseline

- currently only 6 layers, 5 out of which are modality specific, 1 is shared
- we experiment with adding one additional moadality specific layer, and one additional shared layer in another experiment

-> more difficult to align mutliple modalities, than just training one -> add one layer -> motivation for modality specific: after 5 layers information might not be high level enough so that one layer can process the information -> add one additional modality specific -> motivation for shared: after 5 layers information might be high level enough, but capturing modality agnostic information might take more than one layer -> add one additional shared

- added shared layer improves performance slightly, but adds 7 million parameters and 41 minutes to training time
- looking at the improvement in zero-shot, which increases the average Recall from 29.93% to 30.8%, this is not much of an improvement, considering the amount of parameters we add to the model

Model	MSCOCO (5K test set)						Flickr30K (1K test set)					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
FLAVA	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
Data2Vec2	0.02	0.08	0.22	0.01	0.10	0.19	0.02	0.12	0.26	0.02	0.06	0.12
MM-D2V2 (Ours)	4.24	12.12	17.96	1.77	6.54	10.91	1.2	4.88	8.18	0.54	2.52	4.58
MM-D2V2 (Ours)†	31.72	56.78	67.9	12.42	31.05	42.5	7.7	26.18	37.6	4.08	17.01	24.26
MM-D2V2 7_2(Ours)†	32.78	58.34	69.3	12.83	31.85	43.4	8.08	27.92	38.6	4.14	17.5	24.82
MM-D2V2 7(Ours)†	30.24	56.48	67.46	11.96	30.48	41.88	7.36	26.42	36.6	3.7	16.58	23.84

Table 1: Comparison of Zero-shot Image-Text and Text-Image Retrieval of first results with FLAVA and Data2Vec2 papers. Because Data2Vec2 is a unimodal model, we embed each image with the D2V2-Image model and each text with the D2V2-Text model. This yields unusable results, as there has been no incentive for the models to learn a shared representation, as both are unimodal. This is why we had to use both the image and the text model to embed the data.
†: This version has been trained with BEiT-2 as the teacher model, not the D2V2 Image model.

Model	MSCOCO (5K test set)						Flickr30K (1K test set)					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Zero-Shot												
FLAVA	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
CLIP	58.4	81.5	88.1	37.8	62.4	72.2	88.0	98.7	99.4	68.7	90.6	95.2
MM-D2V2 (Ours)	31.72	56.78	67.9	12.42	31.05	42.5	7.7	26.18	37.6	4.08	17.01	24.26
Finetune												
BEiT-3	84.8	96.5	98.3	67.2	87.7	92.8	98	100	100	90.3	98.7	99.5
VLMo	74.8	93.1	96.9	57.2	82.6	89.8	92.3	99.4	99.9	79.3	95.7	97.8

Table 2:

- looking at the validation loss of image and text seperatly, on COCO val set, we observe that the loss on images is significantly lower than the loss on text, which might be due to the fact that the teacher model is a vision model and the target, the cls token, might be biased towards the image modality, as it is unimodal
- interestingly, this bias also seem to be directly translated to the performance on image-text retrieval, as the performance on image-text retrieval is significantly higher than on text-image retrieval -> we are learning the cls token representation, and using the leared cls token as an output for the student model, to encode a modality for retrieval and other downstream task

-> suggests that the cls token is biased towards the image modality, or rather that the model is better in encoding images than text

- we can see that the performance of e.g. BEiT-3 and VLMo is also lower on text-image retrieval than on image-text retrieval, but not the the extend that we observe with our model

Model	MSCOCO (5K test set)		Flickr30K (1K test set)	
	Image → Text	Text → Image	Image → Text	Text → Image
MM-D2V2 7(Ours)†	51.39	28.11	23.46	14.71
BEiT-3	93.2	82.57	99.33	96.17
VLMo	88.27	76.53	97.2	90.93

Table 3: Average recall of image-text and text-image retrieval on MSCOCO and Flickr30K. All models continously perform better on image-text retrieval than on text-image retrieval, but the difference is more pronounced for our model.

- currently vl layer(s) (or rather mulimodal layer(s)) are randomly initialized, one option is to specifically initialize the multimodal layers with the weight of the final layers of the D2V text model -> initial state is closer closer to text modality -> did not work
- currently embedding layer of text and patch embed layer of image model frozen, includes cls token of text and image frozen
- fuzed layer takes the cls token of the image model and the cls token of the text model as input
- so maybe unfreezing the embedding layers will help

-> did not work

- however, when we disable weight decay for image cls token and for text embedding layer, contains text cls/bos token, then we observed an increase in performance

- for all other params, weight decay stays at 0.01
- test adding image-mm and text-mm projections between encoders and shared encoder (FLAVA)
  - check retrieval performance when using ourputs of encoders, are they already a little bit aligned?
  - check retrieval performance when using outputs of projections, are they more aligned?
  - check retrieval performance when using outputs of shared encoder, are they even more aligned?
  - avg. cosine similarity for positive pairs and negative pairs
- we do not compare with See, Hear, and Read: Deep Aligned Representations use the average median rank instead of recall at a specific percent, and from their experimental setup it is not clear which samples they used from Visual Genome for their retrieval experiments.
- what is interesting however, is that with just model transfer, which is Knowledge-Distillation in our case, their model did not perform well on zero-shot retrieval -> halfed score of linear regression -> especially for image-sound retrieval just model transfer, i.e. labeled variant of KD, did not work well
- what made the important difference, which might also be the case for us, is Contrastive Learning, which, with the exception of BEiT-3, was used by both VLMo, FLAVA as one of the pretraining tasks, and for CLIP it was the only pretraining task

Image-Text Contrastive Learning

- solution is to also incorporate contrastive learning into our training
- as we still do KD, we now have two losses, the KD loss and the contrastive loss
  - nothing unusual, done by VLMo, FLAVA (use masked modality modeling as second pretraining task)
  - only contrastive loss done by CLIP
- how is it done in the papers? -> generally always the same
  - take the cls token of the text encoder output, and the cls token of the image encoder output
  - project each of them into a shared embedding space
  - compute the cosine similarity between the image embeddings and the text embeddings of the current batch
  - projection is done by linear layer, popularized by CLIP -> done the same across VLMo, FLAVA, CLIP, BEiT-3
- VLMo additionally takes the output of cls token of the whole model, not of the text and image encoder, and projects it into the shared embedding space with different linear layers
  - each model usually has two projection layers, one for the image encoder and one for the text encoder
  - VLMo has four projection layers, two additional for the cls token of the VL-expert
    - one if the output of the VL-expert is for text, and one if it is for image
  - is surprising, as the the VL-expert forces to learn a shared representation, so projecting it into a shared space with seperate projection layers seems counterintuitive, more intuitive would be to use one projection layer for the output of the cls token of the VL-expert
  - authors did not provide a reason for this
- therefore, we will start with the following:
  - seperate projection layers for image encoder and text encoder -> used to project image/text into multimodal space (FLAVA)
  - one projection layer for the cls token of the shared layer(s) -> used for contrastive learning
  - for unimodal finetuning: use output of the encoder without projection

    - for multimodal finetuning: use output of the shared layer without projection
    - for retrieval: use output of the projection layer
- why even use shared/fuzed layers, why not directly use the same approach as in CLIP? -> test the following: just train a text model with beit-2

-> (CLIP like) just train a text model to regress cls token output of final beit-2 layer -> use blocks of d2v2 text to init text model (generally one could take any pretrained text model) -> freeze embedding layer + pos embedding layer -> has the advantage that max possible context stays 512 tokens, so no need to interpolate for downstream tasks, even if we now use just 64 tokens

- have seperate projection layer for cls token of text encoder and image encoder
- contrastive loss is done based on the cosine similarity of the projected cls tokens
- output of the cls token of the shared multimodal layer(s) is ignored for now
- also means that we use the projected cls tokens for retrieval, which is now not zero-shot anymore, as we explicitly train the model to maximize the cosine similarity between the cls tokens of an matching image-text pair
    - in that regard what FLAVA claims is not true, as they name their results on cross-modal retrieval for COCO and Flickr30K zero-shot, but pretrain their model using a contrastive loss
- we have 5 encoder layers for each modality, and two shared layers

-> means we now use 5 layers + projection for contrastive learning, and therefore for retrieval -> performance result will be interesting

- the question is in which case we will then utilize the output of the shared layers (cls tokens)
- for multimodal task we would use the output of the projections of the encoder's cls tokens
- for unimodal tasks we would use the output of the encoders without the projections
- therefore better option would be to use the cls token of the last layer output, which is a shared one, and project this into the shared space
- even though the representation should already be shared at this point -> so no projection necessary
- for modality specific tasks -> use output of the corresponding encoder
- for multimodal tasks -> use output of the shared layer

following options:
- no shared layers, seperate image and text encoders and two linear projections to shared space (CLIP)
- shared layers, seperate image and text encoders and one linear projection to contrast space
    - use output of shared layer for multimodal tasks, output of encoder for modality specific tasks
    - output of projection for retrieval

CLIP: no shared layers, seperate image and text encoders and two linear projections to shared space
FLAVA: shared layers, seperate image and text encoders, two linear projections to shared space for image and text encoder, two lin projections for image/text to multimodal space for mm encoder, on unimodal downstream classification tasks: use output of respective encoder withouth projection
- we do not do it exactly the same as in FLAVA, as our shared layers are not (yet) for referencing multimodal tasks, but for aligning the modalities on the "concept" level, so use single projection layer for contrastive learning on this output
- also test exactly as in flava
    - have one projection layer for the final output of the cls token, stemming from the shared layer(s)

Contrastive Learning with Memory Bank

Decaying Memory Bank

Importance of the Teacher Model

- BEiT-2 vs. D2V2 Image shows significant difference in performance
- Model distilled from BEiT-2 teacher outperforms the one from D2V2 Image teacher by a large margin
- teacher model size is around the same -> both use ViT-B/16, BEiT-2 around one percent better on Imagenet-1k after finetuning
- too small of a difference that this could be the reason for the large difference in performance
- most likely the handling of the CLS token, which is regressed by our students, is the reason
  - D2V2 Image introduces special CLS loss to aggregate as much (global) information as possible
    - cls token regresses mean activations of all patches
    - was inspired by BEiT-2
  - BEiT-2 introduces a bottleneck to force the model to push as much information as possible towards the cls token
  - latter seems to be more effective
- which teacher to use does make a difference!