HAL: Improved Text-Image Matching by Mitigating Visual Semantic Hubs

Fangyu Liu,[1*†] **Rongtian Ye,**[2*] **Xun Wang,**[3*] **Shuaipeng Li**[4]
[1]University of Cambridge, Cambridge, UK
[2]Aalto University, Espoo, Finland
[3]Malong Technologies, Shenzhen, China
[4]SenseTime Research, Beijing, China

Abstract

The *hubness problem* widely exists in high-dimensional embedding space and is a fundamental source of error for cross-modal matching tasks. In this work, we study the emergence of hubs in *Visual Semantic Embeddings* (VSE) with application to text-image matching. We analyze the pros and cons of two widely adopted optimization objectives for training VSE and propose a novel hubness-aware loss function (HAL) that addresses previous methods' defects. Unlike (Faghri et al. 2018) which simply takes the hardest sample within a mini-batch, HAL takes all samples into account, using both local and global statistics to scale up the weights of "hubs". We experiment our method with various configurations of model architectures and datasets. The method exhibits exceptionally good robustness and brings consistent improvement on the task of text-image matching across all settings. Specifically, under the same model architectures as (Faghri et al. 2018) and (Lee et al. 2018), by switching only the learning objective, we report a maximum R@1 improvement of 7.4% on MS-COCO and 8.3% on Flickr30k.[1]

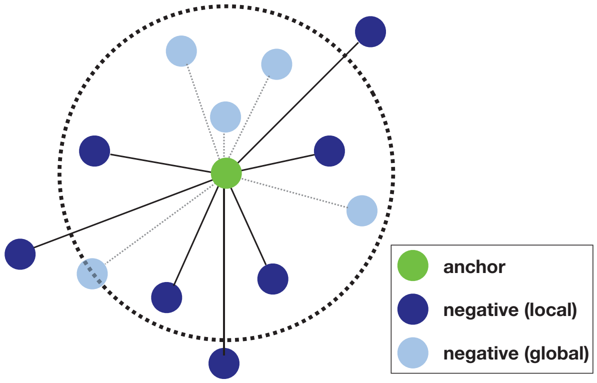

Figure 1: Visualization of our proposed objective, which is to leverage both local and global negative samples to identify hubs in high-dimensional embeddings and learn to avoid them. Local negatives are the ones within mini-batch while global ones are sampled from the whole training set.

Introduction

The hubness problem is a general phenomenon in high-dimensional space where a small set of source vectors, dubbed hubs, appear too frequently in the neighborhood of target vectors (Radovanović, Nanopoulos, and Ivanović 2010). As embedding learning goes deeper, it has been a concern in various contexts including object classification (Tomašev et al. 2011), image feature matching (Jegou et al. 2008) in Computer Vision and word embedding evaluation (Schnabel et al. 2015; Faruqui et al. 2016), word translation (Dinu, Lazaridou, and Baroni 2015; Lazaridou, Dinu, and Baroni 2015) in NLP. It is described as "a new aspect of the dimensionality curse" (Bellman 1961; Schnitzer et al. 2012).

In this work, we study the hubness problem in the task of text-image matching. In recent years, deep neural models have gained a significant edge over non-neural methods in cross-modal matching tasks (Wang et al. 2016). Text-image matching has been one of the most popular ones among them. Most deep methods involve two phases: 1) training: two neural encoders (one for image and one for text) are learned end-to-end, mapping texts and images into a joint space, where items (either texts or images) with similar meanings are close to each other; 2) inference: for a query vector in modality A, a nearest neighbor search is performed to match the query vector against all item vectors in modality B. As the embedding space is learned through jointly modeling vision and language, it is often referred as *Visual Semantic Embeddings* (VSE). Recent work on VSE has shown a clear trend of growing dimensions in order to obtain better embedding quality (Wehrmann 2018). With deeper embeddings, visual semantic hubs increase dramatically. Such property is undesired as the data is structured in the form of text-image pairs and a one-to-one mapping firmly exists among all text and image points.

However, the hubness problem is neither well noticed nor well addressed by current methods of training VSE. Since the start of this line of work (Frome et al. 2013;

[*] Equal contributions.
[†] Correspondence to F. Liu <fl399@cam.ac.uk>.

[1] Our code is released at: https://github.com/hardyqr/HAL.

Kiros, Salakhutdinov, and Zemel 2015), VSE models use either sum-margin (Sum, Eq. (2)) or max-margin (Max, Eq. (3)) ranking loss (both are triplet based) to cluster the positive pairs and push away the negative pairs. Sum is robust across various settings but treats all triplets equivalently and utilizes no information from hard samples, thus does not address the hubness problem at all. Max excels at mining hard samples and achieved state-of-the-art on MS-COCO (Faghri et al. 2018). However, it does not explicitly consider the hubness problem, nor does it resist noise well. New models on training VSE have been consistently brought up in recent years. They include incorporating extra knowledge to augment original data, eg. generating adversarial samples (Shi et al. 2018), and designing high-level objective that utilizes pre-trained models to align salient entities across modalities (Lee et al. 2018; Wu et al. 2019). However, ever since (Faghri et al. 2018), the basic scheme of training VSE has not been enhanced. In this work we show that exploiting the data per se has yet reached its limit.

To fully extract the information buried within, we combine robustness with hard sample mining, proposing a self-adjustable hubness-aware loss called Hal. Hal takes both global (sampled from the whole training set) and local statistics (obtained from mini-batch) into account, leveraging information of hubs to automatically adjust weights of samples. It learns from hard samples and is robust to noise at the same time by taking multiple samples into account. Specifically, we exploit a sample's relationship to 1) other samples within the mini-batch; 2) its k-nearest neighbor queries in a memory bank, to decide its weight. The larger a hub is, the more it should contribute to the loss, resulting in a mitigation of hubs and an improvement of embedding quality. Through a thorough empirical comparison, we show that our method outperforms Sum and Max loss on various datasets and architectures by large margins.

The major contribution of this work is a novel training objective (Hal) that utilizes both local and global statistics to identify hubs in high-dimensional embeddings. Compared with strong baselines (Faghri et al. 2018) and (Lee et al. 2018), Hal improves R@1 by a maximum of 7.4% on MS-COCO and 8.3% on Flickr30k.

Method

We first introduce the basic formulation of VSE model; then review widely-adopted methods that we will compare to; in the end, propose our intended loss function.

Basic Formulation

The bidirectional text-image matching framework consists of a text encoder and an image encoder. The text encoder is composed of word embeddings, a GRU (Chung et al. 2014) (or other sequential models) layer and a temporal pooling layer. The image encoder is usually a deep CNN and a linear layer. We use ResNet152 (He et al. 2016), Inception-ResNet-v2 (IRv2) (Szegedy et al. 2017) and VGG19 (Simonyan and Zisserman 2014) pre-trained on ImageNet (Deng et al. 2009) in our models. We denote them as functions f and g, which map text and image to some vectors of size d respectively.

For a text-image pair (t, i), the similarity of t and i is measured by cosine similarity:

$$S_{it} = \left\langle \frac{f(t)}{\|f(t)\|_2}, \frac{g(i)}{\|g(i)\|_2} \right\rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}. \tag{1}$$

During training, a margin based triplet ranking loss is usually adopted to cluster positive pairs and push negative pairs away from each other. There are mainly two prevalent choices which are Sum and Max. We introduce them in the next section along with our newly proposed non-triplet-based loss Hal.

Revisit Two Triplet-based Loss Functions

In this section we review the two popular loss functions that have been adopted for training VSE and analyze their pros and cons.

Sum-margin Loss (Sum). Sum is a standard triplet loss adopted from the metric learning literature and has been used for training VSE since the start of this line of work (Frome et al. 2013; Kiros, Salakhutdinov, and Zemel 2015). Its early form can be found in (Weston, Bengio, and Usunier 2010) which was used for training joint word-image embeddings. Formally, Sum is defined as:

$$\begin{aligned}\mathcal{L}_{\textsc{Sum}} = &\sum_{i\in I} \sum_{\bar{t}\in T\setminus\{t\}} [\, \alpha - S_{it} + S_{i\bar{t}} \,]_+ \\ &+ \sum_{t\in T} \sum_{\bar{i}\in I\setminus\{i\}} [\, \alpha - S_{ti} + S_{t\bar{i}} \,]_+,\end{aligned} \tag{2}$$

where $[\cdot]_+ = \max(0, \cdot)$; α is a preset margin; T and I are text and image encodings in a mini-batch; t is the descriptive text for image i and vice versa; $\bar{t}$ denotes non-descriptive texts for i while $\bar{i}$ denotes non-descriptive images for t.

The major shortcoming of Sum lies in the fact that it views all valid triplets within a mini-batch as equal and assigns identical weights to all, leading to a failure of identifying informative pairs. As we will detail in the following, a simple "hard" weighting by taking only the hardest triplet can greatly enhance a triplet-based loss's performance in training VSE.

Max-margin Loss (Max). Faghri et al. (2018) proposed Max fairly recently (2018). Max differs from Sum by considering only the largest violation of margin within the mini-batch instead of summing over all margins:

$$\begin{aligned}\mathcal{L}_{\textsc{Max}} = &\sum_{i\in I} \max_{\bar{t}\in T\setminus\{t\}} [\, \alpha - S_{it} + S_{i\bar{t}} \,]_+ \\ &+ \sum_{t\in T} \max_{\bar{i}\in I\setminus\{i\}} [\, \alpha - S_{ti} + S_{t\bar{i}} \,]_+.\end{aligned} \tag{3}$$

We refer to Max as a "hard" weighting strategy as it implicitly assigns a weight of 1 to the hardest triplet and 0 to all other triplets. Though Max was not used in the context of VSE before, it was thoroughly exploited in other embedding learning tasks (Wu et al. 2017). As analyzed by (Wu et al.

2017), a rigid stress on hard negatives like MAX makes its gradient easily dominated by noise, being a result of either deficiency of the model architecture or data's structure per se. Through error analysis, we notice that the existence of *pseudo hardest negatives* in training data is a major source of noise for MAX. During training, only the hardest negative in a mini-batch is considered. If that sample contained happens to be incorrectly labeled or inaccurate, misleading gradients would be imposed on the network. Notice that SUM eases such noise in labels by taking all mini-batch's samples into account. When a small set of samples are with false labels, their false gradients would be canceled out by other correct negatives within the mini-batch, preventing the model from an optimization failure or overfitting to incorrect labels. That being said, SUM fails to make use of hard samples and does not address the hubness problem at all. It thus performs poorly on a well-labeled dataset like MS-COCO.

Besides, both SUM and MAX are triplet based, considering only one positive pair and one negative pair at a time. Such sampling manner isolates each triplet and disregards the overall distribution of data points. What's more, the triplet-style heuristics is easy for selected triplets to satisfy after the early stage of training, leaving very little information in gradients in the late stage (Yu et al. 2018). As opposed to triplet loss, our proposed NCA-based loss, to be introduced in the next section, characterizes the whole local neighborhood and take the affinities among all pairs into consideration.

The Hubness-Aware Loss (HAL)

On the one hand, we obtain the greatest possible robustness through considering multiple samples; on the other hand, we try to make sure the samples being considered are hard enough - so that the training is effective. We tackle this problem by leveraging information from visual semantic hubs. Inspired by Neighborhood Component Analysis (NCA) (Goldberger et al. 2005) used for classification task, we propose a self-adaptive Hubness-Aware Loss (HAL) that weights samples within a mini-batch according to both local and global statistics. More specifically, HAL assigns more weights to samples which appear to be hubs (being close neighbors to multiple queries), judging from both the current mini-batch and a memory bank sampled from the whole training set.

How global and local information are used will be detailed shortly. Before that, we briefly explain NCA and discuss why it is a natural choice for addressing hubness problem. In the classification context, NCA is formulated as:

$$\mathcal{L}_{\text{NCA}} = \sum_{i=1}^{N} \left(\log \sum_{y_i=y_j} e^{S_{ij}} - \log \sum_{k=1}^{N} e^{S_{ik}} \right), \tag{4}$$

where N is the number of samples. And the gradient of $\mathcal{L}_{\text{NCA}}$ w.r.t. positive and negative samples are computed as:

$$\begin{aligned} w^{+} &= \left| \frac{\partial \mathcal{L}_{\text{NCA}}}{\partial {S_{ij}}^{+}} \right| = \frac{e^{S_{ij}}}{\sum_{y_i=y_j} e^{S_{ij}}} - \frac{e^{S_{ij}}}{\sum_{k=1}^{N} e^{S_{ik}}}, \\ w^{-} &= \left| \frac{\partial \mathcal{L}_{\text{NCA}}}{\partial {S_{ij}}^{-}} \right| = \frac{e^{S_{ij}}}{\sum_{k=1}^{N} e^{S_{ik}}}. \end{aligned} \tag{5}$$

For a sample S_{ij}, when it is a close neighbor to multiple items in the search space, ie. being a hub, its weight as a positive is reduced and that as a negative is scaled up, meaning that it receives more attention during training. This basic philosophy of NCA will be used in both the local and global weighting schemes in the following.

a) Global weighting through Memory Bank (MB). One of the most desired property of an NCA-based loss is that it automatically assigns weights to all samples in one batch of back-propagation through computing gradients as suggested above. The more data points we have, the more reliable a hub can be identified. The most ideal approach of leveraging hubs is utilizing the idea of NCA and searching for hubs across the whole training set, so that all samples are compared against each other and information is made fully use of. However, it is computationally infeasible to minimize such objective function on a global scale - especially when it comes to computing gradients for all training samples (Wu, Efros, and Yu 2018). We thus design hand-crafted criteria that follows the NCA's idea to explicitly compute weight of samples but does not require gradient computation. Specifically, at the beginning of each epoch, we sample all over training set and compute their embeddings to create a memory bank M that approximates the global distribution of training data. Then we utilize relationships among mini-batch and memory bank to compute a global weight for each sample in the batch, highlighting hubs and passing the weight to the next stage of local weighting.

We define a function $\texttt{kNN}(x, M, k)$ to return the k closest points (measured by l_2 distance) in point set M to x and the global weighting of HAL can be formulated as:

$$\begin{aligned} W_{ii} = 1 - e^{\alpha(S_{ii}-\epsilon_1)} / \\ \left(e^{\alpha(S_{ii}-\epsilon_1)} + \sum_{\bar{t}\in K_1} e^{\alpha(S_{i\bar{t}}-\epsilon_2)} + \sum_{\bar{j}\in K_2} e^{\alpha(S_{\bar{j}i}-\epsilon_2)} \right), \\ W_{it} = \left(\sum_{\bar{t}\in K_1} e^{\beta(S_{i\bar{t}}-\epsilon_2)} + \sum_{\bar{i}\in K_2} e^{\beta(S_{\bar{i}t}-\epsilon_2)} \right) / \\ \left(e^{\beta(S_{ii}-\epsilon_1)} + e^{\beta(S_{tt}-\epsilon_1)} + \sum_{\bar{t}\in K_1} e^{\beta(S_{i\bar{t}}-\epsilon_2)} + \sum_{\bar{i}\in K_2} e^{\beta(S_{\bar{i}t}-\epsilon_2)} \right), \end{aligned} \tag{6}$$

where W_{ii}, W_{it} represent weight of positive and negative samples respectively; $K_1 = \texttt{kNN}(i, M_T \backslash \{t\}, k), K_2 = \texttt{kNN}(t, M_I \backslash \{i\}, k)$; α, β are temperature scales and ϵ_1, ϵ_2 are margins. For positive weighting, when the anchor's neighborhood is dense, the denominator of the second term gets larger and so does W_{ii}. As will be shown in gradient computation (Eq. (8)), a large W_{ii} scales up positive sample's gradient. Analogously, for negative weighting, a dense neighborhood leads to a large W_{it} and increases the gradient of that negative sample in local weighting.

Table 1: Quantitative results on Flickr30k (Young et al. 2014). “ours” means our own implementation.

#	architecture	loss	image→text					text→image					
			R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r	rsum
1.1	GRU+VGG19	SUM	30.0	59.6	67.7	4.0	34.7	22.8	49.4	61.4	6.0	47.5	291.0
1.2		MAX	30.1	56.3	67.9	4.0	30.5	21.3	47.1	58.7	6.0	40.2	281.4
1.3		HAL	**38.4**	**63.3**	**73.4**	**3.0**	**20.1**	**26.7**	**53.3**	**64.9**	**5.0**	**32.1**	**320.0**
1.4	Order (VGG19, ours) (Vendrov et al. 2016)	SUM	31.4	58.3	69.4	4.0	26.9	24.2	50.9	62.9	5.0	34.3	297.1
1.5		MAX	32.1	58.0	69.9	4.0	23.1	22.7	49.4	61.3	6.0	32.9	293.4
1.6		HAL	**36.4**	**62.2**	**73.0**	**3.0**	**20.4**	**26.6**	**54.4**	**65.6**	**4.0**	**31.0**	**318.3**
1.7	SCAN (Lee et al. 2018)	MAX	67.9	89.0	94.4	-	-	43.9	**74.2**	**82.8**	-	-	452.2
1.8		HAL	**68.6**	**89.9**	**94.7**	1.0	3.3	**46.0**	74.0	82.3	2.0	14.3	**455.5**

b) Local weighting through loss function. Here we adapt the NCA loss for classification for our context of producing a matching among two sets of points:

$$\begin{aligned}\mathcal{L}_{\text{HAL}} = \frac{1}{N}\sum_{i=1}^{N}\Big(&\frac{1}{\gamma}\log(1+\sum_{m\neq i} e^{\gamma W_{mi}(S_{mi}-\epsilon)}) \\ &+\frac{1}{\gamma}\log(1+\sum_{n\neq i} e^{\gamma W_{in}(S_{in}-\epsilon)}) \\ &-\log(1+W_{ii}S_{ii})\Big),\end{aligned} \tag{7}$$

where γ is a temperature scale; ϵ is a margin; N is number of samples within the mini-batch. And the gradients with respect to negative and positive samples are computed as:

$$\begin{aligned} w^{+} &= \left|\frac{\partial \mathcal{L}_{\text{HAL}}}{\partial {S_{ij}}^{+}}\right| = \frac{W_{ij}}{1+W_{ij}S_{ij}} \text{ if } i=j, \\ w^{-} &= \left|\frac{\partial \mathcal{L}_{\text{HAL}}}{\partial {S_{ij}}^{-}}\right| = \underbrace{\frac{W_{ij}e^{\gamma W_{ij}(S_{ij}-\epsilon)}}{1+\sum_{m\neq j} e^{\gamma W_{mj}(S_{mj}-\epsilon)}}}_{\text{weighted by image modality}} \\ &+ \underbrace{\frac{W_{ij}e^{\gamma W_{ij}(S_{ij}-\epsilon)}}{1+\sum_{n\neq i} e^{\gamma W_{in}(S_{in}-\epsilon)}}}_{\text{weighted by text modality}}. \end{aligned} \tag{8}$$

Unlike a naive NCA aiming for classifying samples in only one direction, the first and second term of $\mathcal{L}_{\text{HAL}}$ punish mistakes made during searching targets among the two modalities in both directions. As shown in gradients, the sample is weighted according to its significance as a hub in both modalities.

HAL VS MAX. As pointed out by (Lazaridou, Dinu, and Baroni 2015), MAX actually implicitly mitigates the hubness problem by targeting the hardest triplet only. A hub, by definition, is a close (potentially nearest) neighbor to multiple queries and would thus be punished by MAX for multiple times (in different batches). (Lazaridou, Dinu, and Baroni 2015)’s experiments also verified such theory empirically. However, it is a risky choice as the hardest sample within a mini-batch can easily be a pseudo hardest negative as analyzed above. As we would show in experiments, HAL prevails in a broader range of data and model configurations while MAX only performs well on some specific circumstances where both training data and encoders are of ideal quality. Also, HAL is essentially leveraging more information than MAX. In MAX, only hub that violates margin the most gets to impose a gradient on network’s parameters while HAL softly considers all hubs, big or small, by assigning them weights.

Experiments

This section is divided into 1) Experimental Setups and 2) Main Results, where detailed configurations of experiments are introduced in 1) and comparison & analysis of main results are in 2).

Experimental Setups

Dataset. We use MS-COCO (Lin et al. 2014) and Flickr30k (Young et al. 2014) as our experimental datasets. For MS-COCO, there have been several different splitting protocols being used in the community. We use the same split as (Karpathy and Fei-Fei 2015): 113,287 images for training, 5,000 for validation and 5,000 for testing.[2] During testing, scores are computed as the average of 5 folds of 1k images. As many of the previous works report test results on a 1k test set (a subset of the 5k one), we would experiment with both protocols. We refer to the 1k test set as $c1$ and the 5k test set as $c2$. Flickr30k has 30,000 images for training; 1,000 for validation; 1,000 for testing.

Evaluation metrics. We use R@Ks (recall at K), Med r, Mean r and rsum to evaluate the results. R@K: the ratio of “# of queries that the ground-truth item is ranked in top K” to “total # of queries” (we use $K \in \{1, 5, 10\}$); Med r: the median of the ground-truth ranking; Mean r: the mean of the ground-truth ranking; rsum: the sum of R@$\{1, 5, 10\}$ for both text→image and image→text. R@Ks and rsum are the higher the better while Med r and Mean r are the lower the better. We compute all metrics for both text→image and image→text retrieval. During training, we follow the convention of taking the model with the maximum rsum on validation set as the best model for testing.

Model and training details. We use 300-d word embeddings and 1024 internal states for GRU text encoder (all randomly initialized with Xavier init. (Glorot and Bengio

[2] Note that 1 image in MS-COCO and Flickr30k has 5 captions, so 5 text-image pairs are used for every image.

2010)); all image encodings are obtained from image encoders pre-trained on ImageNet (for fair comparison, we don't finetune any image encoders); $d = 1024$ for both text and image embeddings. For more details about hyperparameters and training configurations please refer to Table 3 and code release: https://github.com/hardyqr/HAL.

Main Results

Here we present the major quantitative and qualitative findings with analysis regarding HAL's performance, hyperparameters' choice and hubs' distributions.

Comparing HAL, SUM and MAX. Table 1 and 2 present our quantitative results on Flickr30k and MS-COCO respectively. On Flickr30k, we experiment three models and HAL achieves significantly better performance than MAX and SUM on the first two configurations.[3] On MS-COCO, HAL also beats both triplet loss functions. Interestingly, while MAX fails badly on Flickr30k, it becomes very competitive on MS-COCO. This serves as an evidence of MAX easily overfitting to small datasets.[4] In conclusion, HAL maintains its edge over MAX and SUM across regardless of data and architecture configurations. Even without global weighting (memory bank), HAL still beats the two triplet losses by a large margin. The equipment of memory bank can usually further boosts rsum by another $3-5$. Also, it is worth noticing that HAL converges significantly faster than MAX and SUM. HAL stabilizes after approximately 5 epochs while MAX and SUM take roughly 10 epochs.

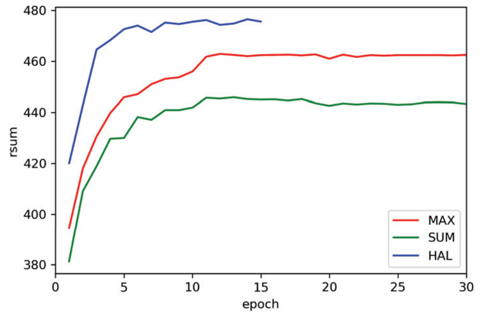

Figure 2: Plotting epoch against rsum on validation set for comparing convergence time. All models are using GRU+ResNet152, trained & validated on MS-COCO.

HAL vs. State-of-the-art. Table 2 line 2.13-2.24 list quantitative results of both our proposed method (2.23, 2.24) and numbers reported in previous works (2.13-2.22). For fair comparison with (Faghri et al. 2018), we only use routine encoder architectures (GRU+ResNet152). Unlike (Shi et al. 2018; Wu et al. 2019), we also do not bring in any extra information to help training. With a trivial configuration of model & data, our method is still ahead of the state-of-the-art on MS-COCO (Wu et al. 2019) by a decent margin for most metrics. Notice that we are comparing against works that use frozen image encoder (as we do). For the ones that finetuning image features, better performance is achievable (Song and Soleymani 2019; Shuster et al. 2019). In Table 2 line 2.25, 2.26, we list SCAN (Lee et al. 2018) alone as it incorporates additional knowledge, i.e. bottom-up attention information, from a Faster R-CNN (Ren et al. 2015) to refine the visual-semantic alignment. With such prior, it is a well-established state-of-the-art on the Text-Image matching task, having much higher rsums than previous works. For SCAN, we pick configurations with the best rsums on both MS-COCO and Flikr30k, switching its learning objective from MAX to HAL.[5] On Flikr30k, MAX and HAL deliver comparable results. On MS-COCO, HAL is significantly stronger - rsum is further improved to **512.7** with R@1 improved by 7.4 and 3.7 for image→text and text→image respectively. We did not experiment with MB due to GPU memory limits.

The impact of batch size. In contrast to loss functions that treat each sample equivalently, batch size does matter to HAL as it defines the neighborhood size where relative similarity is considered during local weighting. And HAL does benefit from a larger batch size as it means an expanded neighborhood. As suggested in Figure 2, on MS-COCO, HAL reaches a maximum rsum with a batch size of 512. Note that in the NCA context, batch size is a relative concept. For Flickr30k, which is only of roughly $\frac{1}{4}$ the size of MS-COCO, we maintain the original batch size of 128 to cover roughly the same range of neighborhood.

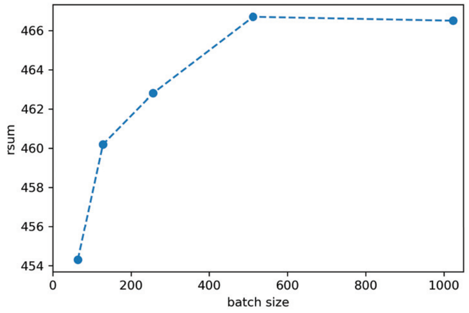

Figure 3: Plotting batch size used by HAL against rsum. All models are using GRU+ResNet152, trained & tested on MS-COCO $c2$.

The impact of size of memory bank. The MB in HAL has two hyperparameters: 1) k, which characterizes the scope of neighborhood being considered for global statistics, and 2) memory bank's size. Their relative scales matter for mining informative samples in the top-k neighborhood. When k is fixed, we search the most appropriate memory

[3] We do not include HAL+MB for (Vendrov et al. 2016) as it demands GPU memory exceeding 11GB, which is the limit of our used GTX 2080Ti. Same reason applies to SCAN+HAL+MB.

[4] (Faghri et al. 2018) showed that data augmentation techniques like random crop applied on input images can improve MAX's performance over small datasets.

[5] An ensemble model is able to achieve even higher rsum but for clear comparison we do not discuss the ensemble case.

Table 2: Quantitative results on MS-COCO (Lin et al. 2014). First three blocks (line 2.1-2.12) are using protocol $c2$ (5k test set); the last two blocks (line 2.13-2.24) is using $c1$ (1k test set) in convenience of comparing with results reported in previous works. Mb means memory bank.

#	**architecture**	**loss**	**image→text**					**text→image**					
			R@1	R@5	R@10	Med r	Mean r	R@1	R@5	R@10	Med r	Mean r	rsum
2.1	GRU+VGG19	Sum	46.9	79.7	89.5	2.0	5.9	37.0	73.1	85.3	**2.0**	11.1	411.5
2.2		Max	51.8	82.1	90.5	**1.0**	5.1	39.0	73.9	84.7	**2.0**	12.0	421.9
2.3		Hal	55.5	84.3	92.3	**1.0**	4.2	**41.9**	75.6	86.7	**2.0**	7.8	436.1
2.4		Hal+Mb	**56.7**	**84.9**	**93.0**	**1.0**	**4.0**	**41.9**	**75.9**	**87.1**	**2.0**	**7.2**	**439.5**
2.5	GRU+IRv2	Sum	50.9	82.7	92.2	1.4	4.1	39.5	75.8	87.2	**2.0**	9.4	428.3
2.6		Max	57.0	86.2	93.8	**1.0**	3.5	43.3	77.9	87.9	**2.0**	8.6	446.0
2.7		Hal	60.2	87.3	94.4	**1.0**	3.3	44.8	78.2	88.3	**2.0**	7.7	453.2
2.8		Hal+Mb	**62.7**	**88.0**	**94.6**	**1.0**	**3.1**	**45.3**	**78.8**	**89.0**	**2.0**	**6.3**	**458.5**
2.9	GRU+ResNet152	Sum	53.2	85.0	93.0	**1.0**	3.9	41.9	77.2	88.0	**2.0**	8.7	438.3
2.10		Max	58.7	88.2	94.8	**1.0**	3.2	45.0	78.9	88.6	**2.0**	8.6	454.2
2.11		Hal	**64.4**	89.2	94.9	**1.0**	3.0	46.3	78.8	88.3	**2.0**	7.9	462.0
2.12		Hal+Mb	64.0	**89.9**	**95.7**	**1.0**	**2.8**	**46.9**	**80.4**	**89.9**	**2.0**	**6.1**	**466.7**
2.13	(Kiros, Salakhutdinov, and Zemel 2015) (ours)		49.9	79.4	90.1	2.0	5.2	37.3	74.3	85.9	**2.0**	10.8	416.8
2.14	(Vendrov et al. 2016)		46.7	-	88.9	2.0	5.7	37.9	-	85.9	**2.0**	8.1	-
2.15	(Huang, Wang, and Wang 2017)		53.2	83.1	91.5	**1.0**	-	40.7	75.8	87.4	**2.0**	-	431.8
2.16	(Liu et al. 2017)		56.4	85.3	91.5	-	-	43.9	78.1	88.6	-	-	443.8
2.17	(You, Zhang, and Luo 2018)		56.3	84.4	92.2	**1.0**	-	45.7	81.2	90.6	**2.0**	-	450.4
2.18	(Wehrmann 2018) (d=1024)		57.8	87.9	95.6	**1.0**	3.3	44.2	80.4	90.7	**2.0**	**5.4**	456.6
2.19	(Faghri et al. 2018)		58.3	86.1	93.3	**1.0**	-	43.6	77.6	87.8	**2.0**	-	446.7
2.20	(Faghri et al. 2018) (ours)		60.5	89.6	94.9	**1.0**	3.1	46.1	79.5	88.7	**2.0**	8.5	459.3
2.21	(Liu and Ye 2019)		58.3	89.2	95.4	**1.0**	3.1	45.0	80.4	89.6	**2.0**	7.2	457.9
2.22	(Wu et al. 2019)		64.3	89.2	94.8	**1.0**	-	48.3	81.7	**91.2**	**2.0**	-	469.5
2.23	GRU+ResNet152 + Hal		65.4	90.4	96.4	**1.0**	2.5	47.4	80.6	89.0	**2.0**	7.3	469.2
2.24	GRU+ResNet152 + Hal + Mb		**66.3**	**91.7**	**97.0**	**1.0**	**2.4**	**48.7**	**82.1**	90.8	**2.0**	5.6	**476.6**
2.25	(Lee et al. 2018) (t-i AVG)		70.9	94.5	97.8	-	-	56.4	**87.0**	**93.9**	-	-	500.5
2.26	(Lee et al. 2018) (t-i AVG) + Hal		**78.3**	**96.3**	**98.5**	1.0	2.6	**60.1**	86.7	92.8	1.0	5.8	**512.7**

bank size and find that 5% of training data is ideal as suggested in Figure 4. The top-k neighborhood of a too large memory bank might be filled with noisy samples (potentially being incorrectly labeled).

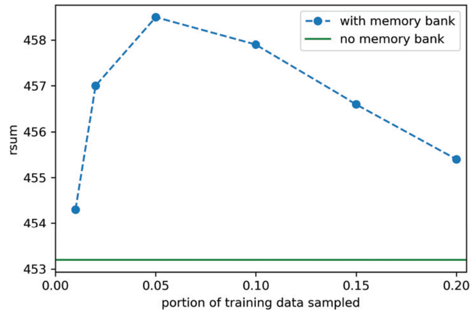

Figure 4: Plotting rsum against Hal's memory bank size. Hal without memory bank is also provided as a baseline. All data points are produced with GRU+IRv2 as the base model and are trained & tested on MS-COCO $c2$.

Related Work

In this section, we introduce works from three fields that are highly-related to our work: 1) text-image matching and VSE; 2) deep metric learning; 3) tackling the hubness problem in various contexts.

Text-image Matching and VSE. Since the dawn of deep learning, works have emerged using a two-branch architecture to connect language and vision. Weston, Bengio, and Usunier (2010) trained a *shallow* neural network to map word-image pairs into a joint space for image annotation. In 2013, Frome et al. (2013) brought up the term VSE and trained joint embeddings for sentence-image pairs. Later works extended VSE for the task of text-image matching (Hodosh, Young, and Hockenmaier 2013; Kiros, Salakhutdinov, and Zemel 2015; Gong et al. 2014; Vendrov et al. 2016; Hubert Tsai, Huang, and Salakhutdinov 2017; Faghri et al. 2018; Wang et al. 2019a), which is also our task of interest. Notice that text-image matching is different from generating novel captions for images (Lebret, Pinheiro, and Collobert 2015; Karpathy and Fei-Fei 2015) but is to retrieve existing descriptive texts or images in a database.

While many of these works improve model architectures for training VSE, few have tackled the shortcomings in learning objectives. Faghri et al. (2018) made the latest attempt to reform the long being used Sum loss. Their proposed Max loss is indeed a much stronger baseline than Sum in most data and model configurations. But it fails significantly when the dataset is small or noise is contained. Liu and Ye (2019) eased such deficiency by relaxing Max into a top-K triplet loss. Shekhar et al.; Shi et al. (2017;

Table 3: Experiment configurations.

#	Datasets	models	hyperparameters
3.1	MS-COCO	2.1, 2.5, 2.9, 2.13	margin=0.2, lr=0.001, lr_update=10, bs=128, epoch=30
3.2		2.2, 2.6, 2.10, 2.20	margin=0.2, lr=0.0002, lr_update=10, bs=128, epoch=30
3.3		2.11, 2.23	γ=30, ϵ=0.3, lr=0.001, lr_update=10, bs=512, epoch=15
3.4		2.12, 2.24	γ=30, ϵ=0.3, α=40, β=40, ϵ_1=0.2, ϵ_2=0.1 lr=0.001, lr_update=10, bs=512, epoch=15
3.5		2.26	γ=100, ϵ=1.0, lr=0.0005, lr_update=10, bs=256, epoch=20
3.6	Flickr30k	1.1, 1.4	margin=0.05, lr=0.001, lr_update=10, bs=128, epoch=30
3.7		1.2, 1.5	margin=0.05, lr=0.0002, lr_update=15, bs=128, epoch=30
3.8		1.3	γ=60, ϵ=0.7, lr=0.001, lr_update=10, bs=128, epoch=15
3.9		1.8	γ=70, ϵ=0.6, lr=0.0005, lr_update=10, bs=128, epoch=30

2018) raised similar concerns. They mainly focused on creating better training data while we target the training objective itself.

Deep Metric Learning. Text-image matching is an open-set task where matching results are identified based on similarity of pairs, instead of assigning probabilities to specific labels in a closed set. Such property coincides with the idea of metric learning, which utilizes relative similarities among pairs to cluster samples of same class in embedding space. Entering the deep learning age, deep neural net based metric learning is widely applied in various tasks including image retrieval (Oh Song et al. 2016; Wang et al. 2019b), face recognition (Schroff, Kalenichenko, and Philbin 2015), person re-identification (Yi et al. 2014), etc.. We use kindred philosophy in our context of matching two sets of data points. Works on deep metric learning that inspired our model are discussed here.

Neighborhood Component Analysis (NCA) (Goldberger et al. 2005) introduced the foundational philosophy for metric learning where a stochastic variant of K-Nearest-Neighbor score is directly maximized. (Yi et al. 2014; Oh Song et al. 2016; Sohn 2016; Wang et al. 2019b) further developed the idea, leveraging the gradient of NCA-based loss to discriminatively learn from samples of different importance. (Wu, Efros, and Yu 2018) proposed a method that computes only part of NCA-based loss's gradient, so that NCA on a large scale is computationally feasible.

Tackling the Hubness Problem. We have stated what the hubness problem is in the introduction. Now we introduce several efforts tackling the hubness problem in various contexts. (Zhang, Xiang, and Gong 2017) pointed out the wide existence of hubs in text-image embeddings but did not address them. Though not receiving enough attention in VSE literature, hubness problem has recently been extensively explored in Bilingual Lexicon Induction (BLI). BLI is the task of inducing word translations from monolingual corpora in two languages (Irvine and Callison-Burch 2017). In terms of finding correspondence between two sets of vectors, it is analogous to our task of interest. (Smith et al. 2017; Lample et al. 2018) proposed to first conduct a direct Procrustes Analysis and then use criteria that heavily punish hubs during inference to reduce the hubness problem. While it is indeed efficient in finding a better matching, the actual quality of embedding is not improved. Joulin et al. (2018) integrated the inference criterion CSLS from (Lample et al. 2018) into a least-square loss and trained a transformation matrix end-to-end to mitigate hubness problem. Though this work has a similar philosophy to ours, it is specifically designed for BLI and only trains one linear layer over two sets of word vectors. When CSLS is appended to a triplet loss, it is merely a resampling of hard samples, making it non-special in terms of both form and intuition.

Conclusion

We introduce a novel loss HAL for mitigating visual semantic hubs during training text-image matching models. The self-adaptive loss HAL leverages the inherit nature of Neighborhood Component Analysis (NCA) to identify information of hubs, from both a global and local perspective, giving considerations to robustness and hard sample mining at the same time. Our method beats two prevalent triplet-based objectives across different datasets and model architectures by large margins. Though our methods have only experimented on the task of text-image matching, there remains to be other cross-modal mapping tasks requiring obtaining a matching, e.g. content-based image retrieval, document retrieval, document semantic relevance, Bilingual Lexicon Induction, etc.. HAL can presumably be used in such settings as well.

Acknowledgments

We thank anonymous reviewers for their careful feedbacks, based on which we were able to enhance the work. We thank our family members for unconditionally supporting our independent research. The author Fangyu Liu gives special thanks to 1) Prof. Lili Mou, who voluntarily spent time reading and discussing the rough ideas with him at the very beginning; 2) his aunt Qiu Wang who supplied him GPU machines; 3) his labmates Yi Zhu and Qianchu Liu from Language Technology Lab for proofreading the camera-ready version.

References

Bellman, R. 1961. Adaptive control processes: a guided tour princeton university press. *Princeton, New Jersey, USA*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 248–255. IEEE.

Dinu, G.; Lazaridou, A.; and Baroni, M. 2015. Improving zero-shot learning by mitigating the hubness problem. *ICLR worshop*.

Faghri, F.; Fleet, D. J.; Kiros, J. R.; and Fidler, S. 2018. Vse++: Improving visual-semantic embeddings with hard negatives.

Faruqui, M.; Tsvetkov, Y.; Rastogi, P.; and Dyer, C. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 30–35.

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; and Mikolov, T. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*, 2121–2129.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Goldberger, J.; Hinton, G. E.; Roweis, S. T.; and Salakhutdinov, R. R. 2005. Neighbourhood components analysis. In *Advances in neural information processing systems*, 513–520.

Gong, Y.; Wang, L.; Hodosh, M.; Hockenmaier, J.; and Lazebnik, S. 2014. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision (ECCV)*, 529–545. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hodosh, M.; Young, P.; and Hockenmaier, J. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47:853–899.

Huang, Y.; Wang, W.; and Wang, L. 2017. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2310–2318.

Hubert Tsai, Y.-H.; Huang, L.-K.; and Salakhutdinov, R. 2017. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3571–3580.

Irvine, A., and Callison-Burch, C. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics* 43(2):273–310.

Jegou, H.; Schmid, C.; Harzallah, H.; and Verbeek, J. 2008. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1):2–11.

Joulin, A.; Bojanowski, P.; Mikolov, T.; and Grave, E. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *EMNLP (short paper)*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Kiros, R.; Salakhutdinov, R.; and Zemel, R. S. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics (TACL)*.

Lample, G.; Conneau, A.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Lazaridou, A.; Dinu, G.; and Baroni, M. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, 270–280.

Lebret, R.; Pinheiro, P. O.; and Collobert, R. 2015. Phrase-based image captioning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37 (ICML)*, 2085–2094. JMLR. org.

Lee, K.-H.; Chen, X.; Hua, G.; Hu, H.; and He, X. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 201–216.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, 740–755. Springer.

Liu, F., and Ye, R. 2019. A strong and robust baseline for text-image matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 169–176. Florence, Italy: Association for Computational Linguistics.

Liu, Y.; Guo, Y.; Bakker, E. M.; and Lew, M. S. 2017. Learning a recurrent residual fusion network for multimodal matching. In *Proceedings of the IEEE International Conference on Computer Vision*, 4107–4116.

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4004–4012.

Radovanović, M.; Nanopoulos, A.; and Ivanović, M. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11(Sep):2487–2531.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Schnabel, T.; Labutov, I.; Mimno, D.; and Joachims, T. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 298–307.

Schnitzer, D.; Flexer, A.; Schedl, M.; and Widmer, G. 2012. Local and global scaling reduce hubs in space. *Journal of Machine Learning Research* 13(Oct):2871–2902.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Shekhar, R.; Pezzelle, S.; Klimovich, Y.; Herbelot, A.; Nabi, M.; Sangineto, E.; and Bernardi, R. 2017. Foil it! find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 255–265.

Shi, H.; Mao, J.; Xiao, T.; Jiang, Y.; and Sun, J. 2018. Learning visually-grounded semantics from contrastive adversarial samples. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3715–3727.

Shuster, K.; Humeau, S.; Hu, H.; Bordes, A.; and Weston, J. 2019. Engaging image captioning via personality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12516–12526.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, S. L.; Turban, D. H.; Hamblin, S.; and Hammerla, N. Y. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *ICLR*.

Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 1857–1865.

Song, Y., and Soleymani, M. 2019. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1979–1988.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Tomašev, N.; Brehar, R.; Mladenić, D.; and Nedevschi, S. 2011. The influence of hubness on nearest-neighbor methods in object recognition. In *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, 367–374. IEEE.

Vendrov, I.; Kiros, R.; Fidler, S.; and Urtasun, R. 2016. Order-embeddings of images and language. *ICLR*.

Wang, K.; Yin, Q.; Wang, W.; Wu, S.; and Wang, L. 2016. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*.

Wang, L.; Li, Y.; Huang, J.; and Lazebnik, S. 2019a. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(2):394–407.

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019b. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5022–5030.

Wehrmann, Jônatas, B.-R. C. 2018. Bidirectional retrieval made simple. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7718–7726.

Weston, J.; Bengio, S.; and Usunier, N. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning* 81(1):21–35.

Wu, C.-Y.; Manmatha, R.; Smola, A. J.; and Krähenbühl, P. 2017. Sampling matters in deep embedding learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.

Wu, H.; Mao, J.; Zhang, Y.; Jiang, Y.; Li, L.; Sun, W.; and Ma, W.-Y. 2019. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6609–6618.

Wu, Z.; Efros, A. A.; and Yu, S. X. 2018. Improving generalization via scalable neighborhood component analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 685–701.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, 34–39.

You, Q.; Zhang, Z.; and Luo, J. 2018. End-to-end convolutional semantic embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5735–5744.

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2:67–78.

Yu, R.; Dou, Z.; Bai, S.; Zhang, Z.; Xu, Y.; and Bai, X. 2018. Hard-aware point-to-set deep metric for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 188–204.

Zhang, L.; Xiang, T.; and Gong, S. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021–2030.