Language Features Matter: Effective Language Representations for Vision-Language Tasks Supplementary

Andrea Burns Reuben Tan Kate Saenko Stan Sclaroff Bryan A. Plummer
Boston University
{aburns4, rxtan, saenko, sclaroff, bplum}@bu.edu

1. Supplementary Material

1.1. Datasets

Flickr30K [29]. This dataset consists of 32K images obtained from the Flickr website, each of which has been annotated with five descriptive captions. We use the splits of Plummer *et al*. [21], which separate the dataset into 30K/1K/1K train/test/validation images which we use for the image-sentence retrieval and phrase grounding tasks.

MSCOCO [16]. This dataset links 123K images for the training and validation sets (80K/40K images, respectively), each of which is annotated with five descriptive captions. For the image-sentence retrieval experiments, we use the test/validation splits from Wang *et al*. [27], which consists of 1K images for each split, for a total of 2K images, randomly sampled from the validation set. For image captioning experiments, use the splits from Chen *et al*. [3], which reserves 5K images each for validation and testing.

Flickr30K Entities [21]. This dataset augments the Flickr30K dataset with 276K bounding boxes which are linked to noun phrases in the descriptive captions. We use the same splits as the Flickr30K dataset, resulting in 14.5K instances across the 1K images in the test set for the phrase grounding task. Following [21, 23, 27], we use the union of the bounding boxes for the ground truth box of a phrase which is linked to multiple boxes.

ReferIt [12]. This dataset augments the 20K images from the IAPR RC-12 dataset [6] with 120K region descriptions. We split the splits of Hu *et al*. [11], which split the images evenly into train/validation and test sets (10K each), resulting in about 60K instances in each split.

DiDeMo [8]. This dataset consists of just over 10,000 videos, each of which has between 3-5 video segment descriptions. We use the splits provided by Hendricks *et al*. [8], which splits the videos into sets of 8.4K/1K/1K for train/test/validation.

VQA v2 [5]. This dataset augments images from MSCOCO with QA pairs. The training, validation and test image sets contain 83K, 41K, and 81K images, respectively. This constitutes 444K, 214K, and 448K questions for training/validation/testing splits. Each training and validation question has ten answers provided.

1.2. Task Methods

Image-Sentence Retrieval. We use a modified implementation of the Embedding Network [27] provided by the authors in our experiments[1]. This model uses two branches, one for text and one for images, to learn a projection to a shared embedding space where Euclidean distance is used to measure similarity between images and sentences. We use the default parameters and data processing in the author's implementation, except that we compute the visual representation for each image using a 152-layer ResNet [7] which has been trained on ImageNet [4]. Additionally, we use 448x448 crops rather than the 224x224 pixel crops used by Wang *et al*. [27] as done in prior work, *e.g*. [30, 18]. Following [27, 30, 18], we keep the CNN parameters fixed for a fair comparison. By default this model uses an Average Embedding language model. When we use the LSTM language model, we use a hidden state of 512-D. We set regularization coefficient α to be 1e-4 when fine-tuning the Average Embedding and Self-Attention model and 1e-6 for the LSTM model.

Phrase Grounding. To evaluate our word embeddings on this task, we use the implementation of CITE network [19][2]. This model learns a set of embeddings which share some parameters, each of which captures a different concept important for

[1] https://github.com/lwwang/Two_branch_network

[2] https://github.com/BryanPlummer/cite

phrase grounding. Following Plummer *et al*. [20], we use the parameters and feature representation learned from fine-tuning a 101-layer ResNet and Region Proposal Network. This model also uses an Average Embedding language model by default, and we use 256-D hidden state for our LSTM experiments. We set regularization coefficient α to be 1e-5 for both datasets.

Text-to-Clip. When we performed our experiments none of the methods on the DiDeMo dataset which outperform the baseline model of Hendricks *et al*. [8] had publicly available code for the text-to-clip task (*e.g*. [2, 17]). As a result, we used the CITE network for the text-to-clip task since it performed better than the baseline model as well as better than the phrase-region grounding Similarity Network [27] and straightforward adaptations of the R-C3D model [28] in our experiments. We learn $K = 8$ concept embeddings for this dataset and use the VGG [25] features for the visual representation provided by Hendricks *et al*. [8]. We use a 512-D hidden state for our LSTM models, and set regularization coefficient α to 5e-2. This dataset likely required additional regularization when fine-tuning its embeddings due to its relatively small size.

Image Captioning. We use a PyTorch implementation [3] of the Auto-Reconstructor Network (ARNet) architecture [3] provided by the authors. This model builds off of the original Neural Image Captioning (NIC) architecture [26] by adding an additional LSTM to reconstruct previous hidden states. We set the regularization coefficient of the NIC loss, α, to be 5e-2 when fine-tuning the word embeddings. ARNet's additional stacked LSTM takes a current hidden state as input and attempts to generate the previous hidden state. This can be viewed as a "soft" zoneout strategy as the model adaptively learns how to reconstruct the last hidden state at each time step, as opposed to the typical zoneout regularizer which makes a binary choice between previous and current hidden states.

Visual Question Answering. We use the authors' implementation[4] of the End-to-End Module Networks [10] as our VQA model. This network learns to decompose natural language questions into sub-tasks and assembles question-specific deep networks from neural modules to solve its corresponding sub-task. The training process of this model consists of two parts: the cloning expert and the policy search. Since the policy search improves the model by only 0.7% while adding significant training time, we report results only using the cloning expert. We use the default parameters in the implementation and follow the authors' data pre-processing steps. When we include L2 regularization on the word embeddings, we set its weight to be 5e-4. Note that we report results using the VQA v2 dataset, whereas Hu *et al*. [10] reported results on VQA v1.

1.3. Additional Task Methods

Image-Sentence Retrieval. We also report results with the Stacked Cross Attention Network (SCAN) model [15] using the authors' provided implementation[5]. Unlike the Embedding Network, this model uses the top 36 region-level features [1] which have been trained to capture image concepts on the Visual Genome dataset [14]. A similarity score is computed between all combinations of words in a sentence and image regions, and then aggregated using a multi-step attention mechanism to obtain an overall matching score. For each dataset, we use the settings for the best performing single model reported in their paper, *i.e*., *i-t AVG ($\lambda 1$ = 4)* for Flickr30K and *t-i AVG ($\lambda 1$ = 9)* for MSCOCO.

Phrase Grounding. To supplement our results, we experiment with using the implementation of the Query Adaptive R-CNN network [9] from Plummer *et al*. [20]. This model adapts Faster R-CNN [22] to the phrase grounding task. The implementation in Plummer *et al*. updates the VGG network used in the original paper with a 101-layer ResNet, but does not pretrain their model on Visual Genome or use the online hard negative mining [24] as done in the original paper. In addition, Plummer *et al*. also reported better performance by randomly sampling 5 phrases associated with an image for each minibatch rather than using all annotated phrases. We compared this implementation using a VGG network to the grounding performance reported in [9] and found it performed similarly on Flickr30K Entities despite these changes, but using a ResNet backbone as done in our experiments does boost performance by 3-8%.

Text-to-Clip. We provide additional results from the Temporally Grounding Natural Sentence in Video (TGN) [2] model. The TGN model consists of 3 components: the encoder, the interactor and the grounder. Visual and language features are first projected into the same embedding space using the encoder. Next, the interactor computes the frame-by-word interactions using the encoded visual and language features. Finally, based on these interactions, the grounder scores and ranks the temporal segment candidates ending at each frame. We note that these results are obtained from our own implementation of the TGN model as the authors have not released code. In our implementation, we adopt the same hyperparameter values as detailed in [2].

Image Captioning. We provide results for two additional image captioning models: the vanilla show-and-tell Neural Image Captioning model (NIC) of Vinyals *et al*. [26] and the popular Bottom-Up Town-Down (BUTD) model from Anderson *et al*.

[3] https://github.com/chenxinpeng/ARNet

[4] https://github.com/ronghanghu/n2nmn

[5] https://github.com/kuanghuei/SCAN

[1]. We set α = 5e-2 as our L2 regularization coefficient when fine-tuning the word embeddings for both models. We use a PyTorch implementation [6] of the NIC model for this task. This model follows an encoder-decoder paradigm inspired by machine translation, in which the probability of a sentence given an image is maximized. A CNN encodes an image which is then fed into a decoder LSTM to form a natural language sentence. Unlike the results reported in Vinyals *et al.*, we use a single model rather than an ensemble, and use a 152-layer ResNet pretrained on ImageNet as our image encoder.

We also use a PyTorch implementation [7] of the Bottom-Up Top-Down Attention image captioning model. BUTD uses a combination of visual attention mechanisms: bottom-up attention is implemented using Faster R-CNN [22] to generate object region proposals and their respective features, which are then weighted by the top-down attention mechanism. The model also adds an attribute predictor to Faster R-CNN. The language model is implemented with two standard LSTMs, where the first layer serves as top-down attention and the second is the language generator. The attention LSTM takes the previous time step output, mean pooled image features, and previously generated word encoding as input. After a Softmax is applied to the output of the attention LSTM, the weighted visual features are passed to the generator LSTM.

Visual Question Answering. We provide additional VQA results using the Bilinear Attention Networks (BAN) model [13]. The BAN model utilizes adaptive region-level features [1] as the visual input. It extracts joint representations from each pair of visual and word features via low-rank bilinear pooling while computing their bilinear interactions using attention maps. We use the provided implementation [8] in our experiments and adopt the same hyperparameter settings as described in [13].

1.4. Discrepancies with Published Work

If available, we use the authors' publicly available code. Baseline results differ from published values despite this. The best results in [15], [26] are obtained using ensemble methods, but our results use a single model. Although, single model [15] with the five-task multi-task trained GrOVLE + ft is on par with ensemble results.

1.5. Comparison of Word2Vec and GloVe

When initially deciding the set of embeddings to use in our experiments, we did consider GloVe. However, there were insignificant differences between Word2Vec and GloVe results (some shown below). Thus, we didn't include it in the main paper due to space constraints as GloVe is also a dated embedding.

	Image-Sentence Retrieval [27]		Phrase Grounding [19]	
	Flickr30k	MSCOCO	Flickr30k Entities	ReferIt
Method	Mean Recall		Accuracy	
Word2Vec	71.9	79.9	70.94	53.54
GloVe	71.9	80.3	70.11	52.18

Table 1. Preliminary experiments showed GloVe performed similarly to Word2Vec.

[6] https://github.com/yunjey/pytorch-tutorial

[7] https://github.com/ruotianluo/self-critical.pytorch

[8] https://github.com/jnhwkim/ban-vqa

1.6. Image-Sentence Retrieval Extended Pretrained Embedding Metrics

	Embedding Network [27]													
	Flickr30K							MSCOCO						
	Image-to-Sentence			Sentence-to-Image				Image-to-Sentence			Sentence-to-Image			
Method	R@1	R@5	R@10	R@1	R@5	R@10	mR	R@1	R@5	R@10	R@1	R@5	R@10	mR
(a) Training from scratch														
Average Embedding	23.3	48.8	61.9	15.6	35.3	44.3	38.2	55.3	85.7	93.7	43.7	76.7	87.1	73.7
Self-Attention	25.9	53.4	66.2	18.1	45.5	58.8	44.6	59.8	88.7	94.9	45.7	79.5	90.0	76.6
LSTM	45.2	72.2	82.6	29.9	59.0	70.9	60.0	62.8	89.4	94.6	48.1	81.0	89.3	77.5
(b) Word2Vec														
Average Embedding	47.6	75.8	84.3	31.8	62.2	73.2	62.5	57.6	87.2	93.7	44.4	78.8	88.1	75.0
Average Embedding + ft	56.7	84.3	91.4	41.6	72.9	82.1	71.5	62.4	89.1	95.0	50.2	82.2	90.2	78.2
Self-Attention	48.7	76.0	84.5	33.0	64.4	75.2	63.6	58.6	87.4	93.2	45.4	79.7	89.4	75.6
Self-Attention + ft	57.0	84.4	91.4	42.4	73.5	82.8	71.9	64.8	91.2	96.4	51.9	83.1	91.9	79.9
LSTM	50.9	81.4	89.3	38.9	70.2	80.5	68.5	53.8	83.4	92.4	42.0	76.0	87.3	72.5
LSTM + ft	52.1	82.4	89.9	39.6	70.0	79.9	69.0	63.5	89.4	95.0	49.7	81.4	90.3	78.2
(c) FastText														
Average Embedding	53.3	82.7	90.3	39.2	70.1	80.0	69.2	62.0	91.0	96.1	48.8	82.0	91.4	78.5
Average Embedding + ft	**59.4**	**86.8**	**92.0**	42.6	73.7	**83.5**	73.0	**66.6**	91.7	96.6	52.7	**84.4**	92.2	**80.7**
Self-Attention	53.6	81.4	90.0	40.0	71.0	81.0	69.5	63.2	90.7	95.9	48.5	82.3	91.1	78.6
Self-Attention + ft	58.8	85.8	91.8	**44.2**	**74.6**	83.3	**73.1**	65.3	**92.0**	**96.7**	**52.8**	84.2	**92.5**	80.6
LSTM	52.7	83.3	89.9	38.6	70.2	79.9	69.1	57.5	89.7	95.1	47.6	81.4	90.6	76.9
LSTM + ft	52.1	81.4	89.0	39.0	69.9	79.6	68.5	65.3	91.5	97.1	51.6	83.7	91.5	80.1
(d) Sentence-Level														
InferSent	56.4	54.4	91.1	40.7	72.3	82.2	71.2	60.8	90.4	96.1	47.6	77.8	85.5	76.4
BERT	57.9	84.9	91.3	41.3	73.0	82.6	71.8	58.6	89.2	95.8	46.2	76.9	85.4	75.4

Table 2. Image-sentence retrieval results for pretrained embeddings.

1.7. Image-Sentence Retrieval Extended Adapted Embedding Metrics

	Embedding Network [27]													
	Flickr30K							MSCOCO						
	Image-to-Sentence			Sentence-to-Image				Image-to-Sentence			Sentence-to-Image			
Method	R@1	R@5	R@10	R@1	R@5	R@10	mR	R@1	R@5	R@10	R@1	R@5	R@10	mR
(a) Word2Vec + wn														
Average Embedding + ft	57.7	85.3	91.5	42.2	73.2	82.3	72.0	63.6	90.8	95.6	51.1	83.2	91.1	79.2
Self-Attention + ft	57.6	86.2	92.1	42.5	73.3	82.7	72.4	64.0	91.5	96.8	51.4	84.3	91.7	80.0
LSTM + ft	53.5	82.8	89.9	39.3	70.2	80.5	69.3	63.8	90.6	95.7	50.2	82.0	90.9	78.9
(b) GrOVLE														
Average Embedding + ft	57.6	85.1	92.0	42.6	73.6	82.6	72.3	65.2	91.8	96.5	52.1	83.9	92.1	80.2
Self-Attention + ft	56.9	84.2	91.7	43.2	73.9	82.8	72.1	**67.6**	91.4	96.3	52.0	83.7	92.1	80.5
LSTM + ft	54.1	82.7	91.1	39.7	70.2	80.1	69.7	65.0	89.6	95.8	49.7	82.0	90.8	78.8
(c) Visual Word2Vec														
Average Embedding + ft	50.0	79.7	87.0	37.0	68.3	78.6	66.8	61.7	90.6	95.8	50.0	82.7	91.2	78.7
Self-Attention + ft	51.3	82.3	89.5	40.9	69.1	79.9	68.8	61.6	91.4	96.7	50.2	83.1	92.4	79.2
LSTM + ft	50.5	78.3	88.6	36.2	67.7	78.7	66.7	56.2	87.3	94.8	42.5	77.3	87.8	74.5
(d) HGLMM (300-D)														
Average Embedding + ft	56.6	84.2	90.8	41.4	72.0	81.2	71.0	65.5	90.7	96.0	51.5	83.4	91.5	79.8
Self-Attention + ft	56.4	84.7	91.3	42.1	73.3	82.2	71.8	66.2	91.0	96.3	51.8	**84.7**	**92.6**	80.4
LSTM + ft	54.1	82.0	90.2	40.2	70.4	80.2	69.5	61.5	89.9	95.3	48.9	81.5	90.4	77.9
(e) HGLMM (6K-D)														
Average Embedding + ft	60.5	86.4	92.9	43.8	73.9	83.3	73.5	67.2	91.7	**97.5**	**53.0**	84.0	92.2	**80.9**
Self-Attention + ft	**61.6**	**88.4**	**94.5**	**46.4**	**75.7**	**84.1**	**75.1**	65.4	**93.0**	97.4	52.6	83.6	90.6	80.6
LSTM + ft	51.4	80.7	89.4	39.1	68.7	78.6	68.0	65.0	90.7	96.1	51.2	82.8	90.9	79.4

Table 3. Image-sentence retrieval results for adapted embeddings.

1.8. Image-Sentence Retrieval Extended Multi-task Trained GrOVLE Metrics

	Embedding Network [27]													
	Flickr30K							MSCOCO						
	Image-to-Sentence			Sentence-to-Image				Image-to-Sentence			Sentence-to-Image			
Method	R@1	R@5	R@10	R@1	R@5	R@10	mR	R@1	R@5	R@10	R@1	R@5	R@10	mR
GrOVLE w/o multi-task pretraining	47.3	78.9	87.0	33.2	65.1	76.8	64.7	56.3	87.4	94.3	44.5	79.0	88.5	75.0
+ multi-task pretraining w/o target task	49.0	79.7	87.7	35.7	66.2	76.3	65.8	60.8	87.3	94.7	46.7	79.7	89.3	76.4
+ multi-task pretraining w/ target task	51.3	68.7	80.7	36.2	64.3	66.3	66.2	65.5	91.6	96.7	51.2	83.6	91.4	80.2
+ multi-task pretraining w/ target task + ft	**58.2**	**85.8**	**91.9**	**42.1**	**73.8**	**84.0**	**72.6**	**66.8**	**93.4**	**97.9**	**51.8**	**85.0**	**92.8**	**81.3**

Table 4. Image-sentence retrieval results for multi-task trained GrOVLE, created using the original set of task models.

1.9. Image-Sentence Retrieval Additional Model Metrics

	Stacked Cross Attention Network (SCAN) [15]													
	Flickr30K							MSCOCO						
	Image-to-Sentence			Sentence-to-Image				Image-to-Sentence			Sentence-to-Image			
Method	R@1	R@5	R@10	R@1	R@5	R@10	mR	R@1	R@5	R@10	R@1	R@5	R@10	mR
Training from scratch	60.8	86.8	92.0	43.0	72.1	81.9	72.8	69.9	94.3	97.4	56.6	87.1	94.0	83.2
Word2Vec + ft	59.7	83.4	90.9	41.2	70.6	79.8	70.9	71.9	94.1	**98.1**	58.2	**87.8**	93.8	84.0
FastText + ft	60.7	86.8	91.5	42.1	73.0	80.8	72.5	71.4	94.4	97.7	58.0	87.4	93.8	83.8
GrOVLE (w/o multi-task pretraining) + ft	61.0	86.7	92.0	42.2	72.7	81.3	72.7	72.3	94.0	97.9	58.4	87.7	**94.4**	84.1
+ multi-task pretraining w/ target task + ft	**65.8**	**89.8**	**94.2**	**46.8**	**76.2**	**84.5**	**76.2**	**74.4**	**94.8**	97.8	**59.1**	**87.8**	94.2	**84.7**

Table 5. Image-sentence retrieval results with the additional retrieval model for from-stratch, Word2Vec, FastText, GrOVLE, and multi-task trained GrOVLE representations. The multi-task trained GrOVLE was created from the full set of additional models.

1.10. Phrase Grounding Additional Model Metrics

	Query Adaptive R-CNN [9]	
	Flickr30k Entities	ReferIt
Method	Accuracy	
Training from scratch	68.56	50.23
Word2Vec + ft	69.78	52.97
FastText + ft	69.27	53.01
BERT	66.30	51.09
GrOVLE (w/o multi-task pretraining) + ft	70.03	53.88
+ multi-task pretraining w/ target task + ft	**71.08**	**54.10**

Table 6. Phrase grounding results with the additional grounding model for from-stratch, Word2Vec, FastText, BERT, GrOVLE, and multi-task trained GrOVLE representations. The multi-task trained GrOVLE was created from the full set of additional models.

1.11. Text-to-Clip Extended Pretrained Embedding Metrics

	CITE [19]			
Method	R@1	R@5	mIOU	Average
(a) Training from scratch				
Average Embedding	15.53	58.21	25.32	33.02
Self-Attention	15.41	57.85	27.17	33.48
LSTM	14.38	59.02	25.08	32.83
(b) Word2Vec				
Average Embedding	15.91	56.08	26.85	32.95
Average Embedding + ft	15.65	55.00	27.10	32.58
Self-Attention	15.87	55.89	27.90	33.23
Self-Attention + ft	15.81	55.48	**28.48**	33.26
LSTM	**16.27**	57.94	26.97	33.73
LSTM + ft	15.49	59.29	25.04	**33.94**
(c) FastText				
Average Embedding	15.22	56.08	26.06	32.45
Average Embedding + ft	15.69	53.72	26.62	32.01
Self-Attention	15.92	56.14	27.87	33.31
Self-Attention + ft	15.60	55.93	27.99	33.17
LSTM	14.40	**60.21**	24.56	33.06
LSTM + ft	14.80	58.02	24.71	32.51
(d) Sentence-Level				
InferSent	14.33	56.10	25.18	31.87
BERT	14.23	58.76	24.39	32.46

Table 7. Text-to-clip results for pretrained embeddings on DiDeMo.

1.12. Text-to-Clip Extended Adapted Embedding Metrics

	CITE [19]			
Method	R@1	R@5	mIOU	Average
(a) Word2Vec + wn				
Average Embedding + ft	16.05	55.89	27.79	33.24
Self-Attention + ft	16.05	57.73	27.16	33.65
LSTM + ft	16.36	59.81	26.32	34.16
(b) GrOVLE				
Average Embedding + ft	**16.53**	56.05	**28.56**	33.71
Self-Attention + ft	15.60	58.16	25.67	33.14
LSTM + ft	15.79	**61.65**	25.98	34.47
(c) Visual Word2Vec				
Average Embedding + ft	14.05	56.90	24.23	31.73
Self-Attention + ft	14.12	55.23	24.11	31.15
LSTM + ft	14.03	58.52	24.31	32.29
(d) HGLMM (300-D)				
Average Embedding + ft	15.96	54.67	27.24	32.62
Self-Attention + ft	16.23	56.07	28.01	33.44
LSTM + ft	15.89	59.84	25.81	33.85
(e) HGLMM (6K-D)				
Average Embedding + ft	15.43	55.79	26.76	32.66
Self-Attention + ft	15.60	57.82	27.30	33.57
LSTM + ft	16.41	60.86	26.59	**34.62**

Table 8. Text-to-clip results for adapted embeddings on DiDeMo.

1.13. Text-to-Clip Extended Multi-task Trained GrOVLE Metrics

	CITE [19]			
Method	R@1	R@5	mIOU	Average
GrOVLE w/o multi-task pretraining	16.34	**60.84**	26.17	34.45
+ multi-task pretraining w/o target task	16.94	58.90	27.88	34.57
+ multi-task pretraining w/ target task	16.96	59.40	28.09	34.82
+ multi-task pretraining w/ target task + ft	**17.05**	59.84	**28.39**	**35.09**

Table 9. Text-to-clip results for multi-task trained GrOVLE on DiDeMo.

1.14. Text-to-Clip Additional Model Metrics

	Temporal GroundNet (TGN) [2]			
Method	R@1	R@5	mIOU	Average
Training from scratch	**26.26**	**74.33**	31.32	43.97
Word2Vec + ft	25.98	74.11	32.06	44.05
FastText + ft	26.13	74.23	30.53	43.64
GrOVLE (w/o multi-task pretraining) + ft	25.54	73.98	**34.24**	**44.59**
+ multi-task pretraining w/ target task + ft	24.91	73.58	32.37	43.62

Table 10. Text-to-clip results with the additional text-to-clip model for from-stratch, Word2Vec, FastText, GrOVLE, and multi-task trained GrOVLE representations on DiDeMo. The multi-task trained GrOVLE was created from the full set of additional models.

1.15. Image Captioning Extended Pretrained Embedding Metrics

	ARNet [3]		
Method	BLEU-4	CIDER	METEOR
(a) Training from scratch			
LSTM	–	–	–
LSTM + ft	26.7	89.7	24.3
(b) Word2Vec			
LSTM	28.1	92.7	24.7
LSTM + ft	**28.5**	**94.0**	**24.8**
(c) FastText			
LSTM	**28.5**	92.7	24.7
LSTM + ft	28.3	93.2	**24.8**

Table 11. Image captioning results for pretrained embeddings on MSCOCO.

1.16. Image Captioning Extended Adapted Embedding Metrics

	ARNet [3]		
Method	BLEU-4	CIDER	METEOR
(a) Word2Vec + wn			
LSTM + ft	28.6	93.3	**24.9**
(b) GrOVLE			
LSTM + ft	28.3	92.5	24.8
(c) Visual Word2Vec			
LSTM + ft	**28.8**	**94.0**	**24.9**
(c) HGLMM (300-D)			
LSTM + ft	28.7	**94.0**	**24.9**
(c) HGLMM (6K-D)			
LSTM + ft	28.0	92.8	24.7

Table 12. Image captioning results for adapted embeddings on MSCOCO.

1.17. Image Captioning Extended Multi-task Trained GrOVLE Metrics

	ARNet [3]		
Method	BLEU-4	CIDER	METEOR
GrOVLE w/o multi-task pretraining	28.5	92.7	**24.7**
+ multi-task pretraining w/o target task	**28.8**	**93.3**	**24.7**
+ multi-task pretraining w/ target task	28.5	92.7	**24.7**
+ multi-task pretraining w/ target task + ft	28.7	93.2	**24.7**

Table 13. Image captioning results for multi-task trained GrOVLE on MSCOCO.

1.18. Image Captioning Additional Model Metrics

	Neural Image Captioning (NIC) [26]		
Method	BLEU-4	CIDER	METEOR
Training from scratch	18.2	62.5	20.3
Word2Vec + ft	18.7	62.8	20.2
FastText + ft	17.9	61.6	17.9
GrOVLE (w/o multi-task pretraining) + ft	**19.4**	**65.4**	20.6
+ multi-task pretraining w/ target task + ft	**19.4**	65.1	**20.9**

Table 14. Image captioning results with an additional captioning model for from-stratch, Word2Vec, FastText, GrOVLE, and multi-task trained GrOVLE representations on MSCOCO. The multi-task trained GrOVLE was created from the full set of additional models.

	Bottom-Up Top-Down Attention (BUTD) [1]		
Method	BLEU-4	CIDER	METEOR
Training from scratch	35.2	109.8	27.2
Word2Vec + ft	35.1	110.8	27.1
FastText + ft	35.2	110.3	27.1
GrOVLE (w/o multi-task pretraining) + ft	35.1	110.4	27.1
+ multi-task pretraining w/ target task + ft	**35.7**	**111.6**	**27.3**

Table 15. Image captioning results with an additional captioning model for from-stratch, Word2Vec, FastText, GrOVLE, and multi-task trained GrOVLE representations on MSCOCO. The multi-task trained GrOVLE was created from the full set of additional models.

1.19. Visual Question Answering Additional Model Metrics

	Bilinear Attention Network (BAN) [13]
Method	Accuracy
Training from scratch	68.68
Word2Vec + ft	69.91
FastText + ft	69.91
GrOVLE (w/o multi-task pretraining) + ft	69.36
+ multi-task pretraining w/ target task + ft	**69.97**

Table 16. Visual Question Answering results with the additional VQA model for from-stratch, Word2Vec, FastText, GrOVLE, and multi-task trained GrOVLE representations on VQA v2. The multi-task trained GrOVLE was created from the full set of additional models.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2, 3, 8

[2] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *EMNLP*, 2018. 2, 7

[3] Xinpeng Chen, Lin Ma, Wenhao Jiang, Jian Yao, and Wei Liu. Regularizing rnns for caption generation by reconstructing the past with the present. In *arXiv:1803.11439v2*, 2018. 1, 2, 7, 8

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1

[5] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[6] Michael Grubinger, Paul Clough, Henning Mller, and Thomas Deselaers. The IAPR TC-12 benchmark – a new evaluation resource for visual information systems, 2006. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[8] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 1, 2

[9] Ryota Hinami and Shin'ichi Satoh. Discriminative learning of open-vocabulary object retrieval and localization by negative phrase augmentation. In *EMNLP*, 2018. 2, 5

[10] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR, abs/1704.05526*, 3, 2017. 2

[11] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, 2016. 1

[12] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1

[13] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *NeurIPS*, 2018. 3, 8

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 2

[15] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, 2018. 2, 3, 5

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[17] Bingbin Liu, Serena Yeung, Edward Chou, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Temporal modular networks for retrieving complex compositional activities in videos. In *ECCV*, 2018. 2

[18] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017. 1

[19] Bryan A. Plummer, Paige Kordas, M. Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *ECCV*, 2018. 1, 3, 6, 7

[20] Bryan A. Plummer, Kevin J. Shih, Yichen Li, Ke Xu, Svetlana Lazebnik, Stan Sclaroff, and Kate Saenko. Revisiting image-language embeddings for open-ended phrase detection. *arXiv:1811.07212*, 2018. 2

[21] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30K Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93, May 2017. 1

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 3

[23] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 1

[24] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 2

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 2

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2, 3, 8

[27] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *arXiv:1704.03470*, 2017. 1, 2, 3, 4, 5

[28] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 2

[29] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014. 1

[30] Ying Zhang and Huchuan Lu. Deep cross-modal projection learning for image-text matching. In *ECCV*, 2018. 1