A Multimodal Recurrent Neural Networks for Generating Novel Sentence Level Image Descriptions

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we present a multimodal recurrent neural network framework. Unlike traditional methods that map both the image features and sentence features into a same space, we directly model the probability distribution of generating a word given previous words. In this way, we can acquire novel image descriptions by sampling from the word probability distribution. We validate the effectiveness of our model on three benchmark datasets (IAPR TC-12 [6], flickr 8K [20], flickr 30K [10]). Our model outperforms the state-of-the-art methods by a large margin. In addition, we show that our model can be used for both image retrieval task and sentence retrieval task, and achieves nontrivial improvement of performance over previous mapping methods which directly optimize the ranking objective function for retrieval.

1 Model Architecture

1.1 Basic recurrent neural network

We briefly introduce the simplest Recurrent Neural Network (RNN) [3, 16, 17] that was widely used for many natural language processing tasks, such as speech recognition. The architecture of the network is shown in Figure 1(a). It has three basic layers: the input word layer $\mathbf{w}$, context layer $\mathbf{s}$ and output layer $\mathbf{y}$. The input, context and output in time t is denoted as $\mathbf{w}(t)$, $\mathbf{s}(t)$, and $\mathbf{y}(t)$. $\mathbf{w}(t)$ is the one-hot representation (a vector that has the same dimension of the vocabulary size and only has one non-zero element) of the current word. We denote $\mathbf{x}(t)$ as a vector that concatenates $\mathbf{w}(t)$ and $\mathbf{s}(t-1)$. They can be calculated as follows:

$$\mathbf{x}(t) = [\mathbf{w}(t)^T \;\; \mathbf{s}(t-1)^T]^T; \quad \mathbf{s}(t) = f_1(\mathbf{U}\cdot\mathbf{x}(t)); \quad \mathbf{y}(t) = g_1(\mathbf{V}\cdot\mathbf{s}(t)); \tag{1}$$

where $f_1(.)$ and $g_1(.)$ are element-wised sigmoid and softmax function respectively.

We define two types of “depths” for a RNN architecture. We denote the number of different layers as the *structure depth* (e.g. the structure depth of the basic RNN shown in Figure 1(a) is 3). The second depth is denoted as the *temporal depth*. Although the basic RNN model is not very deep in terms of the structure depth, it is a very deep structure when we unfolded temporally. Accordingly, when we do the backpropagation, we need to propagate the error through recurrent connections back in time (Backpropagation Through Time [21]).

1.2 Our m-RNN model

The structure of our multimodal Recurrent Neural Network (m-RNN) is shown in Figure 1(b). The m-RNN model is much deeper than the basic RNN model. Its structure depth is 7 (i.e. input word layer, word embedding layer, hidden layer, recurrent layer, multimodal layer, softmax layer and the next work layer).

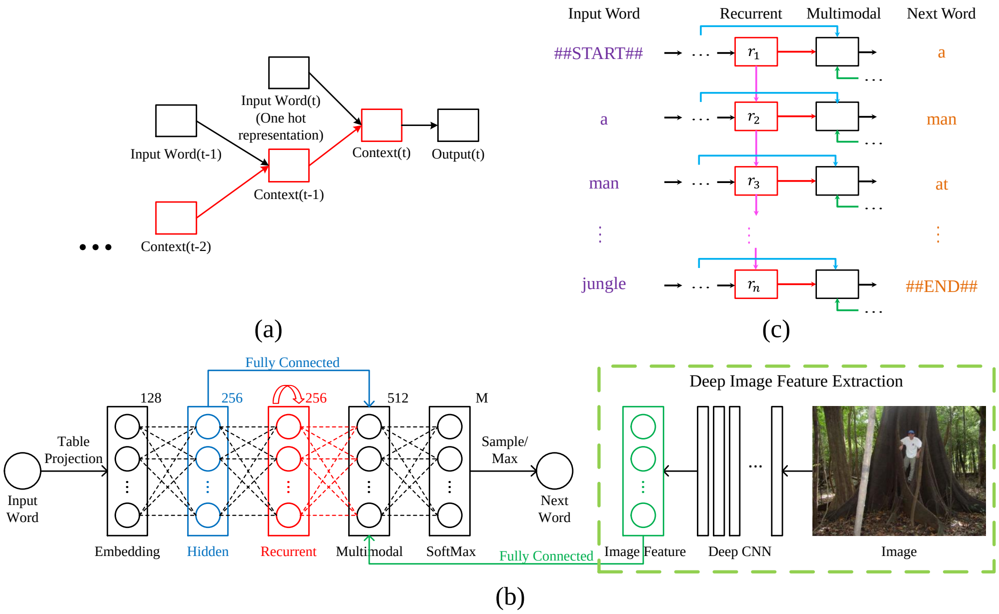


Figure 1: Illustration of the basic Recurrent Neural Network (RNN) and our multimodal Recurrent Neural Network (m-RNN) architecture. (a). The basic RNN architecture. (b). The architecture of m-RNN model. The input of the model is a image and its corresponding sentences. (e.g. the sentence for the shown image is: a man at a giant tree in the jungle. The model will estimate the probability distribution of the next word given previous words and the image. This architecture is deeper than previous widely used RNN in the layer structure sense of depth. (c). The illustration of how the recurrent layer in m-RNN works. We can unfold the recurrent layer, which leads to temporal sense of depth for the network. The model parameters are shared for each temporal session of the unfolded m-RNN model.

The word embedding layer will learn a dense word embedding representation to replace the one-hot representation in basic RNN. It has several advantages. Firstly, it will significantly lower down the number of parameters in the networks because the dense word vector (128 dimension) is much shorter than the one-hot word vector (often larger than 10K dimension, depends on the vocabulary size). Secondly, the dense work embedding will encode the semantic meanings of the words. We can find the semantically neighbored words by calculating the Euclidean distance between two dense word vectors while calculating the distance is meaningless for one-hot representation.

Most of the sentence-image multimodal models [11, 4, 22, 12] uses pre-computed word embedding vectors as the initialization of their model. Different from them, we randomly initialize our word embedding layers and learn them from the training data. We show that this random initialization is enough for our architecture to generate the state-of-the-art results. To future refine the word dense vector representation, we add a hidden layer after the initial word embedding layer and treat the feature vector of this layer as the final word representation.

After the hidden layer, we have a recurrent layer with 256 dimensions. The calculation of the recurrent layer is also different from the basic RNN structure. Instead of concatenating the word representation in time t and the recurrent layer vector in time $t-1$, we first map the word representation in time t and context in time $t-1$ into a same vector space and add them together:

$$\mathbf{s}(t) = f_2(\mathbf{U}_w \cdot \mathbf{w}(t) + \mathbf{U}_s \cdot \mathbf{s}(t-1)); \tag{2}$$

where $\mathbf{s}$ and $\mathbf{w}$ denotes the recurrent layer vector and the word representation respectively. This strategy will reduce the number of parameters and accelerate the training and testing process.

Inspired by the recent success of the Rectified Linear Unit (ReLU) in training very deep structure in computer vision field [13], we replace the element-wised sigmoid function in basic RNN with ReLU function. ReLU is faster, and less easier to be saturated or overfitted the data than many non-linear functions, such as sigmoid. Previous methods [16, 17] to conduct backpropagation through time (BPTT) [21] for RNN suffers from the vanishing gradient problem because even the simplest the RNN model has large temporal depth. They need to use some heuristics, such as truncated BPTT, to avoid this problem. Truncated BPTT will stop the BPTT after k time steps, where k is a hand-defined hyperparameter. Because of the good properties of ReLU, we do not need to stop the BPTT at early stage, which leads to a better and more efficient utilization of the data than truncated BPTT.

After the recurrent layer, we set up a 512 dimensional multimodal layer that connect the language model part and the image part of the whole m-RNN model. The language model part includes the hidden layer (the word representation) and the recurrent layer (the language context). The image part contains the image features. Here we connect the seventh layer of AlexNet [13] to the multimodal layer (please refer to Section 3 for more details). But our framework is open to any image feature learning networks or handcrafted image features. We map the feature vector for each layer to a same feature space and add them together to obtain the feature vector for the multimodal layer:

$$\mathbf{m}(t) = g_2(\mathbf{V}_w \cdot \mathbf{w}(t) + \mathbf{V}_s \cdot \mathbf{s}(t) + \mathbf{V}_I \cdot \mathbf{I}); \tag{3}$$

where $\mathbf{w}$ denotes the multimodal layer feature vector, $\mathbf{I}$ denotes the image feature, $g_2(.)$ is the element-wised scaled hyperbolic tangent function:

$$g_2(x) = 1.7159 \cdot \tanh(\frac{2}{3}x) \tag{4}$$

This function will force the gradients into the most non-linear value range and is faster to learn the parameters of the network than the basic hyperbolic tangent function [14].

As the basic RNN, our m-RNN model has a softmax layer that will generate the probability distribution of next word. The dimension of this layer is the vocabulary size M, which is different for different datasets.

2 Training Objective

We use the average logarithm of the *Perplexity* of the sentences in the training set given their corresponding images as the cost function of our m-RNN model. Perplexity is a standard approach to evaluating language model. The perplexity for one word sequence (i.e. a sentences) $w_{1:L}$ can be calculated as follows, L is the length of the word sequences:

$$\log_2 \mathcal{PPL}(w_{1:L}|\mathbf{I}) = -\frac{1}{L}\sum_{n=1}^{L} \log_2 P(w_n|w_{1:n-1}, \mathbf{I}) \tag{5}$$

where $\mathcal{PPL}(w_{1:L}|\mathbf{I})$ denotes the perplexity of the sentence $w_{1:L}$ given the image $\mathbf{I}$. $P(w_n|w_{1:n-1}, \mathbf{I})$ is the probability of the current word w_n given $\mathbf{I}$ and previous words $w_{1:n-1}$. It corresponds to the feature vector of the SoftMax layer of our modal.

The cost function of our model is the average of the logarithm of the perplexity for all the sentences in the training set:

$$\mathcal{C} = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \mathcal{PPL}(w_{1:L}^{(i)}|\mathbf{I}^{(i)}) \tag{6}$$

where N is the number of sentences in the training set. It is equivalent to the reciprocal of the geometric mean of the probability for the model to generate the training sentences. Our training objective is to minimize this cost function, which is equivalent to maximize the probability of the model to generate the sentences in the training set given their corresponding images. The cost function is derivable thus we can use backpropagation to learn the model parameters.

3 Learning of Image and Sentence Features

The architecture of our model allows the gradients from the loss function to be backpropagated to both the language modeling part (i.e. the word embedding, hidden and the recurrent layers) as well as the image part (e.g. the AlexNet [13]).

For the language modeling part, as mentioned above, we randomly initialize the language modeling layers and learn their parameters. For the image part, we connect the seventh layer of a pre-trained Convolutional Neural Network [13, 2] (aso denoted as AlexNet). The same features extract from the seventh layer of AlexNet (also denoted as decaf features [2]) was widely used by previous multimodal methods [12, 4, 11, 22]. In a most recent work [11], the same image features combined with the state-of-the-art detection framework of Region-CNN [5] were used. Their experiments showed that using this detection framework will indeed increase the performance. In the experiments, we will show that our method performs much better than [11] when the same image features are used, and even better than their results of more sophisticated detection features in terms of many evaluation metrics.

We can update the AlexNet according to the gradient backpropagated from the multimodal layer. In this paper, we fix the image features and the deep CNN network in the training stage due to the luck of the data (The datasets we used in the experiment have less than 30K images). In the future work, we will try our method on large dataset and finetune the parameters of the deep CNN network in the training stage.

4 Sentence Generation, Image and Sentence Retrieval

After training of the m-RNN model, we can use the model for the tasks of sentence generation, image retrieval using sentences and sentence retrieval using images.

The sentence generation process is straight forward. Start from the start sign "##START##" or any length of the context words (e.g. we can give the first words in the reference sentences), our model can calculate the probability distribution of the next word: $P(w|w_{1:n-1}, \mathbf{I})$. Then we can sample from this probability distribution to pick the next word. In practice, we find that picking the word with the maximum probability will perform slightly better than sampling. After that, we input the picked word to the model and sample the next word. This process repeated until we have the end sign "##END##".

For the retrieval tasks, we can use our model to calculate the perplexity of generating a sentence given an image. The perplexity can be treated as an affinity measurement between sentences and images. For the image retrieval task, we just need to retrieve the images that generate the minimum perplexity with the sentence query.

The sentence retrieval tasks is trickier because there might be some sentences that has high probability to any image query. Instead of looking at the perplexity or the probability of the sentences given the query image, we need to use the normalized probability for each sentence: $P(w_{1:L}|\mathbf{I})/\sum_{\mathbf{I}'} P(w_{1:L}|\mathbf{I}^{'})$ where $\mathbf{I}^{'}$ are images sampled from the training set, $P(w_{1:L}|\mathbf{I}) = \mathcal{PPL}(w_{1:L}|\mathbf{I})^{-L}$.

5 Experiments

5.1 Datasets

We test our method on three benchmark datasets with sentence level annotations: IAPR TC-12 [6], flickr 8K [20], and flickr 30K [10].

Here are some statistics and our experimental settings for the three datasets:

IAPR TC-12 Benchmark This dataset consists of around 20,000 images taken from locations around the world. This includes images of different sports and actions, people, animals, cities, landscapes, and so on. For each image, they provide at least one sentences annotations. On average, there are about 1.7 sentences annotations for one image. We adopt the publicly available separation of training and testing set as previous works [7, 12]. There are 17,665 images for training and 1962 images for testing.

Flickr8K Benchmark This dataset consists of 8,000 images extracted from Flickr. For each image, they provide five sentences annotations. The grammar for the annotations of this dataset is simpler than those of the IAPR TC-12 dataset. We adopt the standard separation of training, validation and testing set which is provided by the dataset. There are 6,000 images for training, 1,000 images for validation and 1,000 images for testing.

Flickr30K Benchmark This dataset is a recent extension of Flickr8K. It consists of 158,915 crowd-sourced captions describing 31,783 images. So for each image, they also provide five sentences annotations. The grammar and style for the annotations of this dataset is similar to Flickr8K. We follow the previous work [11] which used 1,000 images for testing. This dataset, as well as the Flick8K dataset, is mainly used for the image-sentence retrieval tasks and there is not public available results of methods for generating novel sentence descriptions.

5.2 Experiments settings

Our model can be used for three tasks: 1) Sentences generation; 2) Sentence retrieval (retrieval top relevant sentences to the given image); 3) Image retrieval (retrieval top relevant images to the given sentence);

5.2.1 Evaluation metrics for sentence generation

Following previous works, we use sentence perplexity and BLEU score [19, 15] as the evaluation metrics. BLEU score was originally designed for automatically machine translation where the task is to give a score to a translated sentences given several references sentences. We can treat the sentence generation task as the ”translation” of the content of image to sentences. The drawback of using BLEU in our task is that for some images, the reference sentences might not contains all the elements and content in the image and the BLEU score might penalize the arguably correct generated sentences, though it remains as the standard evaluation metric for sentence generation methods for image. To conduct a fair comparison, we adopt the same sentence generation steps and experiment settings as [12], and generate as many words as there are in the reference sentences. Note that our model actually do not need to know the length of the reference sentence because we add a end sign ”##END##” at the end of every training sentences and we can stop the generation process when our model outputs the word ”##END##”.

5.2.2 Evaluation metrics for sentence retrieval and image retrieval

For Flickr8K and Flickr30K datasets, we adopted the same evaluation metrics as previous works [22, 4, 11] for both the tasks of sentences retrieval and image retrieval. They used R@K (K = 1, 5, 10), which is the recall rate of the first groundtruth sentences (sentence retrieval task) or images (image retrieval task) as the measurements. Higher R@K usually means better retrieval performance of different methods. Since we care most of the top retrieved results, the R@K with smaller K is more important than those with larger K. In addition to R@K, they used the Med r, which is the median rank of the first groundtruth sentences (sentence retrieval task) or images (image retrieval task). Lower Med r usually means better performance.

For IAPR TC-12 datasets, we adopt exactly the same evualation metrics as [12], which plotted the mean number of matches of the retrieved groundtruth sentences or images with respect to the percentage of the retrieved sentences or images for the testing set. For sentences retrieval task, [12] used a shortlist of 100 images which are the nearest neighbors of specific testing image in the image feature space. This shortlist makes the task harder because similar images might have similar descriptions and it is often harder to find the subtle difference among the sentences and pick the most suitable one. Although there is no published R@K score and Med r score for this dataset as the best of our knowledge, we also report these metrics of our method for future comparison.

5.3 Results on IAPR TC-12

The results of generated sentences is shown in Table 1. BACK-OFF GT2 and GT3 are n-grams methods with Katz backoff and Good-Turing discounting [1, 12]. Ours-RNN-Base has the same architecture with our m-RNN model except that we will not input the image features to the network. It serves as a baseline for our m-RNN model.

To conduct a fair comparison, we followed the same experimental settings of [12], includes the context length to calculate the BLEU score and perplexity. Please note that the perplexity is calculated

	PERP	B-1	B-2	B-3
BACK-OFF GT2	54.5	0.323	0.145	0.059
BACK-OFF GT3	55.6	0.312	0.131	0.059
LBL [18]	20.1	0.327	0.144	0.068
MLBL-B-DeCAF [12]	24.7	0.373	**0.187**	0.098
MLBL-F-DeCAF [12]	21.8	0.361	0.176	0.092
Gupta et al. [9]	/	0.15	0.06	0.01
Gupta & Mannem [8]	/	0.33	0.18	0.07
Ours-RNN-Base	7.77	0.3134	0.1168	0.0803
Ours-m-RNN	**6.92**	**0.3951**	0.1828	**0.1311**

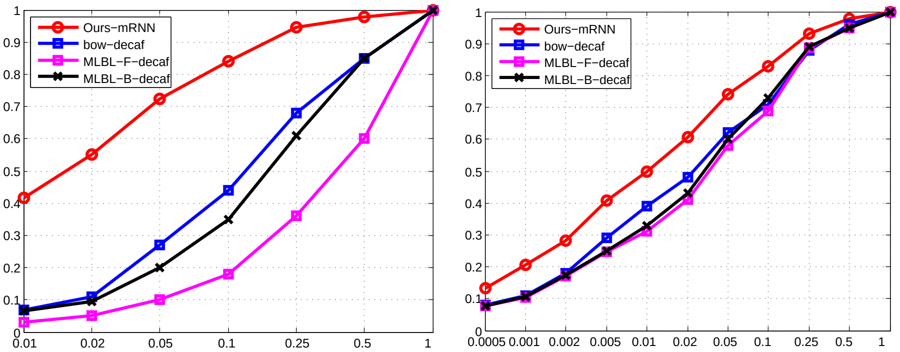
Table 1: Results of generated sentences in the iaprtc-12 dataset.

according to the conditional probability of the words given all of its previous reference words in the sentences. Therefore, a strong language model that successfully captures the grammar of sentences can have a low perplexity without the image content. Perplexity does not directly correlate to the BLEU score where we need to sample the words from the probability distribution generated by the model. For example, although for this dataset, our baseline method of RNN can generate a very low perplexity, it failed to generated sentences with high quality since its BLEU score is not very high. From this perspective, the BLEU score is a better measurement for the generating sentences.

From the experiments, we can see that our m-RNN model performs much better than our baseline RNN model in terms of both perplexity and BLEU score. It also outperforms the state-of-the-art methods in terms of perplexity, B-1, B-3, and a comparable result for B-2.

For retrieval tasks, as mentioned in Section 5.2.2, we draw a recall accuracy curve with respect to the percentage of retrieved images (Text to Image) or retrieved sentences (Image to Text) shown in Figure 2. For sentence retrieval task, we used a shortlist of 100 images as the three comparing methods [12]. The first method, bowdecaf, is a strong image based bag-of-words baseline. The second and the third models are all multimodal deep models. Our m-RNN model outperforms these three methods by a large margin.

Since there are no publicly available results of R@K and median rank in this dataset, we report R@K scores of our method in table 2 for future comparisons.



(a) Image to Text Curve

(b) Text to Image Curve

Figure 2: Retrieval recall curve for (a). Sentence retrieval task (Image to Text) (b). Image retrieval task (Text to Image) of iaprtc-12 dataset.

	Sentence Retrival (Image to Text)				Image Retrival (Text to Image)			
	R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
Ours-m-RNN	20.9	43.8	54.4	8	13.2	31.2	40.8	21

Table 2: Results of R@K and median rank (Med r) for iaprtc-12 dataset.

5.4 Results on flickr8K

This dataset was widely used for a benchmark dataset of Text to Image retrieval and Image to Text retrieval. There are no publicly available methods that reports the statistics of generated sentences.

We first show the R@K evaluation metric in Table 3. Our model outperforms the state-of-the-art methods (i.e Socher-decaf, DeViSE-decaf, DeepFE-decaf) by a large margin when using the same image features (i.e. decaf features). In a recent work [11], they showed that using more sophisticated image features (e.g. decaf feature combined with detection results), will increase the performance. We also list the result of the methods using such features in Talbe 3. Socher-avg-rcnn and DeViSE-avg-rcnn used features of the average CNN activation of all objects above a detection confidence threshold [11]. DeepFE-rcnn used a image feature that further utilizes the RCNN detection algorithm. Socher-avg-rcnn and DeepFE-rcnn all shows better results than their original version of Socher-decaf and DeepFE-decaf. From the table, we can see that our method even performs better than these methods in most of the evaluation metrics. We will develop our framework using the detection algorithm in the future work.

	Sentence Retrival (Image to Text)				Image Retrival (Text to Image)			
	R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
Random	0.1	0.5	1.0	631	0.1	0.5	1.0	500
Socher-decaf [22]	4.5	18.0	28.6	32	6.1	18.5	29.0	29
Socher-avg-rcnn [22]	6.0	22.7	34.0	23	6.6	21.6	31.7	25
DeViSE-avg-rcnn [4]	4.8	16.5	27.3	28	5.9	20.1	29.6	29
DeepFE-decaf [11]	5.9	19.2	27.3	34	5.2	17.6	26.5	32
DeepFE-rcnn [11]	12.6	32.9	44.0	14	9.7	29.6	**42.5**	**15**
Ours-m-RNN-decaf	**14.5**	**37.2**	**48.5**	**11**	**11.5**	**31.0**	42.4	**15**

Table 3: Results of R@K and median rank (Med r) for flickr8K dataset.

We also reports the results of generated sentences in Table 4. There is no publicly available algorithm that reported results on this dataset. We compared our m-RNN model with the Ours-RNN-Base model. The m-RNN model performs much better than the baseline both in terms of the perplexity and BLEU scores.

	PERP	B-1	B-2	B-3
Ours-RNN-Base	30.39	0.4383	0.1849	0.1339
Ours-m-RNN	**24.39**	**0.5778**	**0.2751**	**0.2307**

Table 4: Results of generated sentences in the flickr8K dataset.

5.5 Results on flickr30K

This dataset is a new dataset and there are only a few methods report their retrieval results on it. We first show the R@K evaluation metric in Table 5. Our method outperforms the state-of-the-art methods in most of the evaluation metrics.

In addition, no publicly available methods reported the results of the sentence generation task as the best of our knowledge. We report the results of generated sentences in Table 6 with a comparison of our RNN baseline.

	Sentence Retrival (Image to Text)				Image Retrival (Text to Image)			
	R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
Random	0.1	0.6	1.1	631	0.1	0.5	1.0	500
DeViSE-avg-rcnn [4]	4.8	16.5	27.3	28	5.9	20.1	29.6	29
DeepFE-rcnn [11]	16.4	**40.2**	**54.7**	**8**	10.3	**31.4**	**44.5**	**13**
Ours-m-RNN-decaf	**18.4**	**40.2**	50.9	10	**12.6**	31.2	41.5	16

Table 5: Results of R@K and median rank (Med r) for flickr30K dataset.

	PERP	B-1	B-2	B-3
Ours-RNN-Base	43.96	0.4699	0.1964	0.1252
Ours-m-RNN	**35.11**	**0.5479**	**0.2392**	**0.1952**

Table 6: Results of generated sentences in the flickr8K dataset.

References

[1] S. F. Chen and R. Rosenfeld. A survey of smoothing techniques for me models. *TSAP*, 8(1):37–50, 2000.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[3] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[4] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[6] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, pages 13–23, 2006.

[7] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*, pages 634–647, 2010.

[8] A. Gupta and P. Mannem. From image annotation to image description. In *Neural information processing*, pages 196–204. Springer, 2012.

[9] A. Gupta, Y. Verma, and C. Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012.

[10] P. Y. A. L. M. Hodosh and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions.

[11] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *arXiv:1406.5679*, 2014.

[12] R. Kiros, R. Zemel, and R. Salakhutdinov. Multimodal neural language models. In *ICML*, 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[14] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[15] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, page 605, 2004.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.

[17] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[18] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *ICML*, pages 641–648. ACM, 2007.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

[20] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.

[22] R. Socher, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. In *TACL*, 2014.