Structured data

19/03/2019

Show and tell: A neural image caption generator

Authors:
Damien Grasset
Alice Guichenez

Professors:
Florence d'Alché-Buc
Slim Essid
Zoltan Szabo

Contents

Introduction

Automatic captioning of images constitutes a widely studied domain in the computer vision community. Both the connection it makes between computer vision and natural language processing and its wide range of possible applications, including help for visually impaired people, make it a hot topic in artificial intelligence. In this document, we present and discuss the paper *Show and tell: a neural image caption generator* written by O. Vinyals, A. Toshev, S. Bengio and D. Erhan in 2014 [1] in which the authors study that problem.

This task is particularly challenging because not only does it require to capture the objects contained in an image, to express how these objects relate to each other and their activities, but it also has to express that knowledge in a natural language. The novelty of the authors' approach is that they propose a single joint model for image captioning instead of stitching together existing solutions to those sub-problems like it was previously done. The latter model is based on a convolution neural network (CNN) that encodes an image into a compact representation, followed by a recurrent neural network (RNN) that generates a corresponding sentence.

In this document, we first summarize the analysis and present its main findings. We then discuss the proposed approach. We finally implement the proposed model to the Flickr8k dataset and study the results.

1 Problem contextualization and description of the approach

1.1 A neural and probabilistic framework

The authors' inspiration comes from the literature of machine translation, where the modelling has evolved from achieving a series a separate tasks to using only RNNs, that read source sentences and transform them into fixed-length vector representations which in turn are used in the initial hidden state of a decoder RNN that generates the target sentence. Similarly to machine translation where the input sentence S is transformed into its translation T by maximizing $P(T|S)$, the authors' model takes an image I as input and is trained to maximize the likelihood $P(S|I)$ of producing a target sequence of words $S = \{S1, S2, \dots\}$ where each word S_t comes from a given dictionary that describes the image adequately.

1.2 Model formalism

The authors propose to maximize the probability of the correct description given the image by using the following formulation:

$$\theta^\star = \arg\max_\theta \sum_{(I,S)} \log p(S|I,\theta))$$

where θ are the parameters of the model, I the input image and S its correct transcription. Since the length of S is unbounded because it represents any sentence, the actual modelling is made through the chain rule to model the joint probability over $S_0, \dots, S_N$, where N is the length of S:

$$\log p(S|I,\theta) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \dots, S_{t-1}, \theta).$$

When it comes to training, (S, I) is a training example pair and $\log p(S|I,\theta)$ is optimized using stochastic gradient descent.

$p(S_t|I, S_0, \ldots, S_{t-1}, \theta)$ can be modelled with a RNN where the variable number of words we condition upon up to $t-1$ is expressed by a fixed length hidden state or memory h_t. h_t is updated after seeing a new input x_t by using a non-linear function f: $h_{t+1} = f(h_t, x_t)$.

In practice, the authors choose for f a Long-Short Term Memory (LSTM) net, which has shown good performance on translation. To represent the images and words as inputs x_t, they decide to use an embedding model for the words and a CNN for the images - CNNs have been widely used for image tasks.

LSTM-based Sentence Generator

The LSTM model is based on a memory cell c encoding knowledge at every time step of what inputs have been observed up to this step. The definition of the gates - that control the behaviour of the cells - and cell updates are as follows:

$$\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1}) \text{ (input gate)} \\
f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \text{ (forget gate)} \\
o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1}) \text{ (output gate)} \\
c_t &= f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \\
m_t &= o_t \odot c_t \\
p_{t+1} &= \text{Softmax}(m_t)
\end{aligned}$$

where the sigmoid σ and the hyperbolic tangent h introduce nonlinearity.

For **training**, the LSTM model is trained to predict each word of the sentence after it has seen the image as well as all preceding words as defined by $p(S_t|I, S_0, \ldots, S_{t-1}, \theta)$. The authors present training in the unrolled form, where a copy of the LSTM is created for each word:

$$\begin{aligned}
x_{-1} &= \text{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0, \ldots, N-1\} \\
p_{t+1} &= \text{LSTM}(x_t), \quad t \in \{0, \ldots, N-1\}
\end{aligned}$$

where each word is represented as a one-hot vector S_t of dimension equal to the size of the dictionary. The image and words are mapped to the same space, the image by using a CNN and the words by using word embedding W_e. The image I is only input once, at $t = -1$, to inform the LSTM about the image contents - adding an extra input has yielded empirically weaker results. The loss is defined by $L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t)$ and is minimized w.r.t. all the parameters of the LSTM, the top layer of the image embedder CNN and word embeddings W_e.

For **inference**, the authors propose to use BeamSearch, that works as follows: iteratively consider the set of the k best sentences up to time t as candidates to generate sentences of size $t+1$, and keep only the resulting best k of them. This approximates well $S = \arg\max_{S'} p(S'|I)$.

1.3 Experimental results

To assess the effectiveness of their model, the authors study five datasets consisting of images and sentences describing these images: Pascal VOC 2008, Flickr8k, Flickr30k, MSCOCO and SBU. For all datasets but SBU, each image has been annotated by labelers with 5 sentences that are relatively visual and unbiased. SBU consists of descriptions given by image owners when they uploaded them to Flickr and is therefore more noisy. The Pascal dataset is used for

Figure 1: Scores on the MSCOCO development set

Metric	BLEU-4	METEOR	CIDER
NIC	**27.7**	**23.7**	**85.5**
Random	4.6	9.0	5.1
Nearest Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

Figure 2: BLEU-1 scores. SOTA stands for the current state-of-the-art.

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

testing only.

The authors use several automatic metrics which they compare to subjective scores provided by human raters. Those raters are workers within the framework of an Amazon Mechanical Turk experiment who evaluate generated sentences with a scale from 1 to 41. The level of agreement between workers is 65%, and in case of disagreement, the scores are averaged.

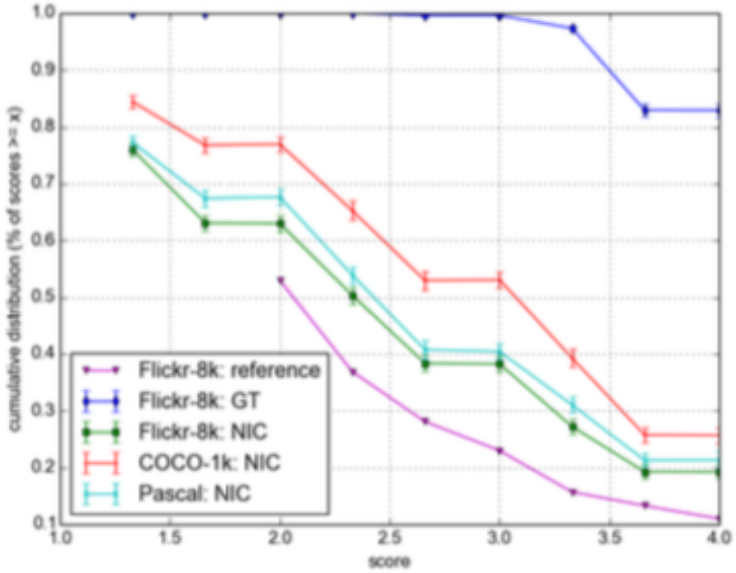
Automatic metrics are computed given groundtruth, i.e. human generated descriptions. The BLEU score, which is a form of precision of word n-grams between generated and reference sentences, has been shown to correlate well with human evaluations.

In terms of training, the authors randomly initialize all weights but the CNN's weights which are initialized to a pretrained model. They train all sets of weights using stochastic gradient descent with fixed learning rate and no momentum. They use 512 dimensions for the embeddings and the size of the LSTM memory. Furthermore, descriptions were preprocessed with basic tokenization, keeping all words that appeared at least 5 times in the training set.

The results are reported in Figures 1 and 2. Since Pascal is only a testing set, they use the system trained using MSCOCO, which is their largest and highest quality dataset. Table 2 shows that their approach yields better results than current state-of-the-art. For Pascal and SBU, the state-of-the-art results do not use image features based on deep learning, so the authors assume that a big improvement on the scores come from that change alone. Human scores were computed by comparing one of the human captions against the other four for the five raters. On the official test set for which labels are only available through the official website, their model has a 27.2 BLEU-4.

Figure 7 shows that NIC is better than the reference system but worse than the groundtruth - expectedly. That shows that BLEU is not a perfect metric, since it does not capture well the difference between NIC and human descriptions assessed by raters.

Figure 3: Scores for the different datasets. GT stands for groundtruth.



2 Discussion of the approach

2.1 Contribution of the paper

In this paper, the authors present an end-to-end system for the problem of automatically describing images. This system is a neural network that has the benefit of being fully trainable using stochastic gradient descent. They take the best out of the state-of-the-art sub-networks for vision and language models to build a better-performing model. The latter indeed yields significantly better performances compared to current state-of-the-art approaches: on the Pascal dataset, their approach yields a BLEU-1 score of 59 as opposed to the current state-of-the-art score of 25, and to be compared to human performance around 69. It also yields BLEU-1 score improvements on Flickr30k, from 56 to 66, and on SBU, from 19 to 28.

As a general rule, one can find in the literature that previous studies have produced highly complex, domain specific and brittle systems that attempted to stitch together solutions to the many sub-problems - capturing the objects contained in the image, expressing how they relate to each other and their activities, and expressing that knowledge in a natural language. The novelty that this paper brings is that it presents a single end-to-end model, combining image classification and natural language sequence generation.

Besides the quantitative accuracy mentioned above, the NIC model shows promising qualitative results as well. First, in regards to generation diversity, the model is able to generate novel captions. When analyzing the N-best list from the beam search decoder instead of the best hypothesis, if one takes the best candidate, the sentence is present in the training set 80% of the times, but half of the top 15 generated sentences are novel. These novel descriptions have a high BLEU socre, which shows that they are of good quality and therefore provide a healthy diversity. Consequently, the model can generate sentences that are diverse and describe different aspects of the same image, and that remain meaningful and valid. Moreover, in regards to the quality of the word embeddings, the model has shown effectiveness in capturing semantic information from the statistics of the language. For instance, the nearest other words found in the learned embedding space for ”street” are ”road”, ”streets”, ”highway”, and ”freeway”. This offers great help for the vision component: having ”horse”, “pony”, and “donkey” close to each

other will encourage the CNN to extract features that are relevant to horse-looking animals. Then, in the extreme case where there are very few examples of a class (e.g. "unicorn"), its proximity to other word embeddings should provide much more information than traditional bag-of-words based approaches.

Furthermore, the choice of an LSTM for f offers a good improvement over a simple RNN in regards to vanishing and exploding gradients. Indeed, passing through t time-steps, the resulting gradient is the product of many gradients and activations. Consequently, gradient messages close to 0 can shrink to 0 while gradient messages larger than 1 can explode. LSTMs - and GRUs - are known to mitigate that problem inherent to RNNs. Indeed, they introduce an additive path between c_t and c_{t-1}, and the gradient climbing prevents gradient explosion. Note that a well chosen activation function is critical, and the hyperbolic tangent, which the authors use, is often recommended.

2.2 Shortcomings

A major issue mentioned in the paper is overfitting. The authors' model is a purely supervised approach and hence requires large amounts of data, but high quality datasets contain less than 100,000 images. The authors explain that the task of assigning a caption to an image is harder than object classification, and data driven approaches have only recently become dominant through datasets as large as ImageNet, which is composed of ten times more data than the datasets in the paper - expect for SBU. Flickr30k is composed of 28,000 images, MSCOCO of 82,783 images and SBU of 1 million images. A consequence of that is that even though the results obtained in the paper are quite good, the advantages of the method over most current human-engineered approaches will only increase once the training set sizes grow. We note that however, the highest BLEU score is not achieved with SBU but with Flickr30k.

Nonetheless, the authors propose several techniques to handle overfitting. First of all, they initialize the weights of the CNN to a pretrained model instead of leaving them uninitialized. They argue that this initialization was of great help in terms of generalization. They also tried to initialize the word embeddings W_e, but they decided to leave them uninitialized as no significant gains were observed. Finally, they used some of the classic overfitting-avoiding techniques, including dropout and ensembling models - which did give a few BLEU points improvement, and they explored the size of the model by trading off number of hidden units versus depth. Note that even though the authors only report BLEU scores because they are usually preferred, model selection and hyperparameter tuning were actually done using perplexity of the model for a given transcription - the geometric mean of the inverse probability for each predicted word. Furthermore, since datasets are limited, one could explore unsupervised approaches to bypass that problem.

Another shortcoming to be noted about the approach is related to transfer learning. Since the authors use five different datasets, they analyze how well they could transfer a model trained from one dataset to another. When training on Flickr30k and testing on Flickr8k, the results are 4 BLEU points higher. In this case, given the proximity between both datasets, the gains are probably due to the additional training data. However, using the very large MSCOCO dataset to train a model tested on the Flickr8k results in lower BLEU scores, despite the fact that this dataset is way bigger than Flick30k. This is likely due to the fact that the collection process and the vocabulary were different. Note that even though all BLEU scores degrade by 10 points, the generated descriptions are still reasonable according to the researchers.

One last point that captured our attention was the choice of the beam size in the use of

Beamsearch. Indeed, the researchers only mention that they tested the values 1 and 20, and that they chose a beam size of 20 given that a beam size of 1 did degrade the results by 2 BLEU points in average. However, the value of 20 seems quite arbitrary with no other justification. During the MS COCO image captioning challenge in 2015, they explored other values to improve their scores and they found, to their surprise, that the best beam size turned out to be small: 3 [2]. Thanks to this further study, the BLUE score was improved by two points. However, as the beam size increases, more candidate sentences are scored and the best one is picked, and consequently increasing the beam size should always yield better sentences. This small beam size as the best size is therefore counter-intuitive, and might indicate that either the model has overfitted or the objective function used to train it - the likelihood - is not aligned with human judgement. The researchers also observed that the novelty of generated sentences was increased when reducing the beam size: training captions are repeated only 60% of the time instead of 80%. This observation supports the fact that the model might have overfitted, and one can then see this reduced beam size technique as another way of regularizing.

2.3 Further topics of discussion

Comparison with baselines

As mentioned above, previous work had only proposed methods stitching together solutions to the different sub-problems. In particular, the authors mention R.Socher, A.Karpathy, Q.V.Le, C.Manning, and A.Y.Ng.'s work [4]. In their paper, they introduce the DT-RNN model which uses dependency trees to embed sentences into a vector space in order to retrieve images that are described by those sentences. This approach, where neural networks are used to co-embed images and sentences together, cannot generate novel descriptions and cannot describe previously unseen compositions of objects, even if the individual objects might have been observed in the training data.

Captioning with attention

K. Xu, J. Lei Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio propose in 2015 [3] an attention-based model founded on the same architecture - CNN encoder and RNN decoder. They base their work on the same end-to-end model but they improve it by incorporating an attention component. They explain that the human visual system is characterized by the presence of attention: instead of compressing an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. Consequently, using representations distilling information in image down to the most salient objects is an effective solution. They explain that a further advantage of including attention is the ability to visualize what the model "sees". They introduce two attention-based image caption generators under the same framework as in [1]: a soft deterministic attention mechanism trainable by standard back-propagation methods, and a hard stochastic attention mechanism trainable by maximizing an approximate variational lower bound. They show how this approach brings valuable insights to interpret the results of this framework by visualizing "where" and "what" the attention focused on. As shown in table 2, the incorporation of attention yields better quantitative results; in particular the BLEU score increases from 63 to 67 for Flickr8k and from 66.6 to 71.8 for COCO.

Model/ Dataset	Flickr8k	Flickr30k	COCO
NIC	63	66.3	66.6
Hard attention	67	66.9	71.8

Table 1: Attention-based model BLEU scores

3 Experimental protocol and implementation

So as to make things a bit more concrete, we tried to use the methodology and the main ideas of this article to build by ourselves an Neural Image Caption Generator. All the code is provided alongside this pdf, and is also available via this link - https://nbviewer.jupyter.org/github/AliceGuichenez/Structured_data/blob/master/Structured-Data-Show-and-Tell.ipynb (where all the details are).

In this section we will explain the experimental choices and steps we used to build it.

3.1 Experimental protocol

Dataset

First of all, we had to make a choice of the image captioning dataset. We chose to use the **Flickr8k** datasets. This dataset contains around 8k images and each of these images is associated with 5 different captions, i.e. sentences that describe the image in English language.

Sentence Processing

Then, we had to process sentences, to make them normalized (lower case for instance) and to build a fixed length vocabulary. To make the size a bit shorter, we only chose words that occur at least 5 times on all the dataset. Plus, we carefully had two new tokens: a <start> token at the beginning of each sentence, and a <end> token at the end. This is essential so that the model knows when a sentence is starting and more importantly, when the sentence is finished which will enable the model to predict the end of a sentence while generating words.

Word Encoder

Once sentences have been processed and the vocabulary fixed, we choose to use a pre-trained embedding to encode each words of our sentences. To do so, we loaded pre-trained weights from the GloVe embeddings (of size 200) and used it as the initialization of the embedding encoding layer of the architecture.

Image Encoder

In order to encode images, we decide to use the pre-trained CNN model Inception V3. To do so, we first normalized images by reshaping them into 224x224x3 shape that we fed into our InceptionV3 model. As this CNN network is originally aimed at classifying images into a large number of classes, we had to remove the top layers (containing the softmax function for instance), so as to obtain vectors of fixed size (2048,). To make the computation and the training a bit faster (especially with a very limited GPU that we have on our computer), we processed all images before and saved the encoded vectors as numpy array on the disk.

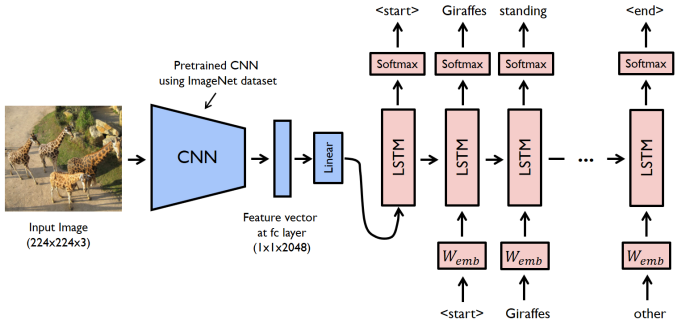
Batch Generator

We constructed our Batch Generator such that it returns at each iteration a batch of images alongside partial captioning sentences in order to predict one word. For a given image i, we iterate over the words of one of his captions s_j where $j \in [1, 5]$, and return for each target word w_t, the partial caption of previous words and the corresponding image $(i, (w_0, ..., w_{t-1}))$

Architecture

We used the following model, the global idea mostly inspired from the article but this architecture have proven to give better results.

Figure 4: Architecture of our Image Caption Generator



As a remark, the summary from Keras tends to mix a bit the order of layers.

The idea of this experimentation was not to get the best results, but more to see and understand the methodology behind this model. Thus we used standard hyper-parameters and optimizer procedure, without trying to fine-tuned them.

1. batch_size = 128, loss = categorical_crossentropy
2. optimizer = Adam, learning_rate = 1e-3 and a ReduceOnPlateau procedure everytime the training got trouble improving the loss.

3.2 Results

We kept a sample of our dataset hidden from the training part in order to test our model on unseen data. We used two different procedure to generate output sentences:

- **Greedy Prediction**: while the model doesn't predict the <end> marker, we keep the word with the highest confidence in the prediction of a word given an image and the past sequence of words. This method can be criticized as we want to predict a whole sentence. In this view, if a word w_t is predict and from this word we predict w_{t+1}, it doesn't mean that the sequence w_t, w_{t+1} is the most likely bigram at time t.

- **Beam Search** with parameter k: to avoid this issues, we can use the beam search method. The idea is to keep at each step of time, the k^{th} most likely sequences of words. To do so, at each time step t, we look at the k^{th} most likely words for each of the k^{th} possible sequences we saved from step $t-1$. Thus it will lead to $k * k$ new sequences and we'll keep the k^{th} most likely new sentences among those k^2 choices.

Here are two examples of captions generated on images of the test set:

Figure 5: Two test pictures

Dog Picture Prediction using Beam Seach (k=5) : 'a white dog is jumping into the water .'
Football Player Prediction using Beam Seach (k=5) : 'a football player in a sooners jersey is running on the ice .'

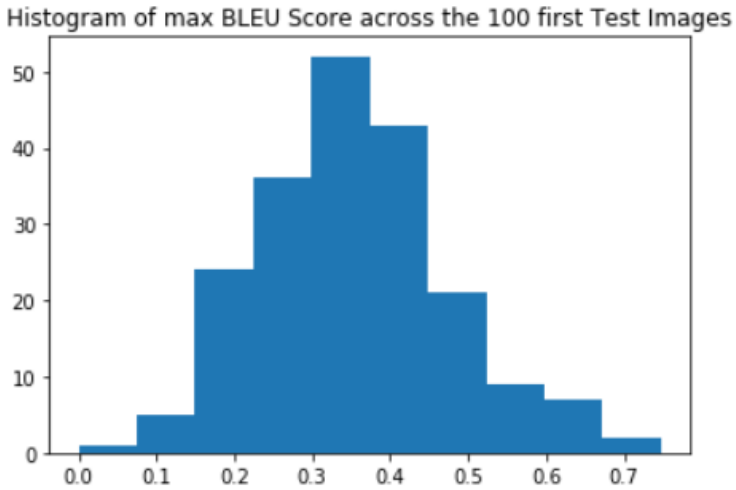
Ground Truth Caption	BLEU Score
A brown dog jumping off a rock into a lake	0.40
A brown dog leaps into water from a rock	0.36
A dog is taking a dive into a body of water	0.45
A dog leaps over the water from a rock	0.38
The dog is leaping into the water	**0.72**
A man is wearing a Sooners red football shirt and helmet	0.46
A Oklahoma Sooners football player wearing his jersey number 28	**0.59**
A Sooners football player wears the number 28 and black armbands	0.52
Guy in red and white football uniform	0.26
The American footballer is wearing a red and white strip	0.36

Table 2: BLEU scores of Ground Truth Caption the predicted sentence

As we can see, sentences generated makes at least sense and are not completely out of context regarding those two (random) test images. Furthermore it is interesting to notice, especially for the dog example, that the model generated a sentence very close to at least one of the ground truth caption (here with a BLEU score of 0.72).

In the Notebook, we evaluated the BLEU score on the 200 first Test Images and kept the best score between the generated sentence and all the ground truth sentences. The idea is that the best match with the generated sentence would be closest best caption that was generated by the model. We only need one of the ground truth to be close are there is no one absolute way to describe the image. Here is the histogram of it :

Figure 6: Two test pictures



For instance we can see that, in some case it is not that good, for instance with the worst BLEU score evaluated on the test Images :

Prediction : a little girl in a pink shirt is standing on a stone wall

Figure 7: Lowest BLEU Score

Conclusion

In this paper, the authors present an end-to-end system for automatically captioning images based on a CNN encoding the image and an RNN generating the sentence. On top of being less complex and more general than the previously existing models, this system has a high quantitative accuracy as well as promising qualitative properties. However, one major drawback of the model is overfitting, but the authors propose several approaches to bypass it. Our implementation of the model also shows good results.

An interesting extension of the paper to investigate would be the incorporation of attention within the model, since it yields better results by imitating the attention component of the human visual system.

References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. *Show and tell: A neural image caption generator*. arXiv:1411.4555, November 2014.

[2] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. *Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge*. 2015.

[3] K. Xu, J. Lei Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio. *Show, attend and tell: neural image caption generation and visual attention*. 2015 (ICML2015)

[4] R.Socher, A.Karpathy, Q.V.Le, C.Manning, and A.Y.Ng. *Grounded compositional semantics for finding and describing images with sentences*. In ACL, 2014.