Gated Recurrent Capsules for Visual Word Embeddings

Danny Francis, Benoit Huet, and Bernard Merialdo

EURECOM, 450 route des Chappes, 06410 Biot, France
firstname.lastname@eurecom.fr

Abstract. The caption retrieval task can be defined as follows: given a set of images I and a set of describing sentences S, for each image i in I we ought to find the sentence in S that best describes i. The most commonly applied method to solve this problem is to build a multimodal space and to map each image and each sentence to that space, so that they can be compared easily. A non-conventional model called Word2VisualVec has been proposed recently: instead of mapping images and sentences to a multimodal space, they mapped sentences directly to a space of visual features. Advances in the computation of visual features let us infer that such an approach is promising. In this paper, we propose a new Recurrent Neural Network model following that unconventional approach based on Gated Recurrent Capsules (GRCs), designed as an extension of Gated Recurrent Units (GRUs). We show that GRCs outperform GRUs on the caption retrieval task. We also state that GRCs present a great potential for other applications.

Keywords: multimodal embeddings · deep learning · capsule networks.

1 Introduction

This paper proposes a novel deep network architecture for the caption retrieval task: given a set of images and a set of sentences, we build a model that ought to find the closest sentence to an input image. That task is important because retrieving captions in natural language using images implies getting closer to a human understanding of visual scenes. Numerous works have attempted to address that task; most of them are making use of a multimodal space where sentences and images are projected and compared [7, 9, 13, 16]. Word2VisualVec [4, 6] relies on another approach, the authors built a model to project sentences directly in a space of visual features: as the quality of visual features is constantly improving, the authors stated that learning visual sentence embeddings rather than projecting them in a more complicated multimodal space was a promising approach. In this paper, a model following this unconventional approach is proposed.

Projecting images and sentences in the same space, whether multimodal or simply visual, implies that representations of images and sentences as mathematical objects must be derived. Since the recent breakthrough of deep learning, Convolutional Neural Networks (CNNs) have shown compellingly good performances in computer vision tasks. In particular, some of them [15, 10] are able to learn visual features that they use to classify images from a big dataset such as ImageNet [3]. Some works have also shown that these visual features could be successfully used in other tasks with different datasets [24]. In particular, most recent works on caption retrieval have used features coming from a ResNet [10] which had been trained on ImageNet for a classification task [7, 9]. In our work, we will extract features thanks to a ResNet that had been finetuned on MSCOCO [17] by the authors of [7]. Deriving visual sentence representations is the main part of our work. Recurrent Neural Networks (RNNs) such as Long Short-Term Memory units (LSTMs) [12] and Gated Recurrent Units (GRUs) [2] have proved to deliver state-of-the-art results on various language modeling tasks such as translation [21], automatic image captioning [23] or caption retrieval [7]. In the last version of Word2VisualVec [6], the authors showed that concatenating a representation derived by a GRU with a Word2Vec [18] representation and a bag-of-words representation to get a multi-scale sentence representation lead to better results in visual sentence embedding for caption retrieval. However, we argue that using these kinds of representations cannot be optimal: pooling all words together without putting attention on relevant parts of the sentence does not reflect the complexity of images; and the current state-of-the-art model for image and caption retrieval is based on object-detection and cross-attention [16], which corroborates our statement that sentences should be processed in a finer way. Our work aims at proposing a new architecture corresponding to and addressing that issue: how to analyze a sentence so that important visual elements are emphasized?

Our research has been inspired by recent works on capsule networks [20, 11]. This new architecture shows promising results in computer vision. In capsule networks, neurons are replaced by so-called capsules, that take vectors as inputs and output vectors. These output vectors are routed towards subsequent capsules through a predefined routing procedure, that can be seen as an attention mechanism: relevant vectors are routed towards relevant capsules. We think that this principle can be successfully used in Recurrent Neural Networks, and the Gated Recurrent Capsule that we introduce in this paper is a novel architecture, and is to our best knowledge the very first occurrence of recurrent unit using capsules.

Our contributions in this paper are three-fold:

- we introduce Gated Recurrent Capsules (GRCs), a novel RNN architecture which extents conventional GRUs so that information flow focuses on critical data items;
- we propose to address the caption retrieval task using the newly proposed GRCs architecture;
- we demonstrate experimentally that GRC enable higher performance when compared to state of the art Word2VisualVec (employing GRUs) in the MSCOCO caption retrieval task.

Our paper is divided in five sections. Having introduced the extent of the paper in Section 1, we will describe related works in Section 2. In Section 3 we will describe our model for caption retrieval. Section 4 details results obtained by our model. We will conclude the paper in Section 5.

2 Related Work

Several works have been done on building visual-semantic embeddings. Most of them are based on the construction of a multimodal space where sentences and images are projected and compared. In [7], Faghri et al. used a GRU to map sentences to a multimodal space; images were simply mapped to that space through a linear transform. They obtained good results by finetuning the ResNet they used to produce visual features: that is the reason why we used one of their finetuned ResNets to produce visual features in our model. Another more complex model proposed by Gu et al. [9] showed that results could be boosted by the use of two generative models (one generating images and one generating sentences) in addition to a GRU and a ResNet. More recently, [16] has shown that even better performances could be reached by processing images with an object detection model combined with cross-attention instead of deriving global visual features.

Another approach has been proposed recently: instead of mapping images and sentences to a multimodal space, [4, 6] proposed to derive visual features from images and to map directly sentences to the space of visual features. This approach is promising as the quality of visual features is constantly increasing. Moreover, it avoids mapping images to a more complex space. Our work follows that unconventional approach. It has been inspired by recent works on capsule networks [20, 11]. Capsule networks have shown promising results in computer vision. However to our best knowledge they have not been used yet in a recurrent fashion for natural language processing apart from [8]; however, the architecture presented in [8] is using a complex GRUs setup to process and route the information, leading to much more learnable parameters, which is a drawback that our architecture does not have.

3 Visual Sentence Embeddings

Word2VisualVec is a non-conventional approach to caption retrieval, as it maps sentences directly to a visual features space. Our model follows that approach.

3.1 Word2VisualVec

In [4], a first version of Word2VisualVec was proposed. It consisted in applying a multilayer perceptron on vectorized sentences to project these sentences in a space of visual features. Three vectorization methods were discussed in that paper: bag-of-words, word hashing and averaging Word2Vec embeddings.

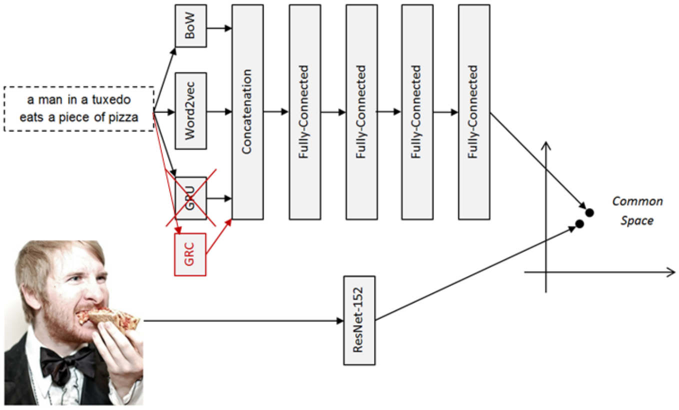


Fig. 1. Word2VisualVec and our variant with a GRC. In Word2VisualVec, three sentence representations (Word2Vec, BoW and GRU) are concatenated and then mapped to a visual features space. In our model, we replaced the final hidden state of the GRU by the average of all final hidden states of a GRC.

In [6], the authors of [4] improved Word2VisualVec by concatenating three sentence representations. In that paper, a sentence representation was produced by concatenating a bag-of-words, a Word2Vec and a GRU representation of the sentence. Then, it was projected in a space of visual features through a multilayer perceptron. Figure 1 shows how Word2VisualVec works in practice.

On top of good performances in caption retrieval, this visual representation of sentences showed an interest in multimodal query composition: the authors showed that visual words features could be added or subtracted to images features and form multimodal queries. Authors also stated that further gains could be expected by including locality in Word2VisualVec representations.

3.2 Gated Recurrent Capsules

Gated Recurrent Units were introduced by Cho et al. in [2]. They are similar to LSTMs: they have similar performances and are well adapted to NLP because they can handle long-term dependencies in sentences. We preferred GRUs to LSTMs because they have less parameters for similar performances. More formally, a GRU is composed of an update gate u_t and a reset gate r_t, and can be described with the following expressions:

$$u_t = \sigma(W_{xu}x_t + W_{hu}h_{t-1} + b_u), \tag{1}$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r), \tag{2}$$

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \circ h_{t-1}) + b_h), \tag{3}$$

$$h_t = (1 - u_t) \circ h_{t-1} + u_t \circ \tilde{h}_t, \tag{4}$$

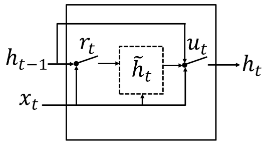


Fig. 2. A Gated Recurrent Unit: for each input x_t, a new value $\tilde{h}_t$ is computed, based on x_t, r_t and h_{t-1}, where r_t expresses how much of h_{t-1} should be reset to compute $\tilde{h}_t$. Eventually, h_t is computed based on $\tilde{h}_t$, h_{t-1} and u_t, where u_t expresses how much of $\tilde{h}_t$ should be used to update h_{t-1} to h_t

with x_t the t-th input and h_t the t-th output or hidden state of the GRU. Here and throughout the paper, $\circ$ denotes the Hadamard product and σ denotes the sigmoid function. The equations above can be explained as follows: for each input x_t, the GRU computes r_t and u_t based on the input and the previous state h_{t-1}. It computes a new value $\tilde{h}_t$ based on x_t, r_t and h_{t-1}, and r_t expresses how much of h_{t-1} should be reset to compute $\tilde{h}_t$. Eventually, h_t is computed based on $\tilde{h}_t$, h_{t-1} and u_t, and u_t expresses how much of $\tilde{h}_t$ should be used to update the hidden state h_t of the GRU. Learned parameters are $(W_{xu}, W_{hu}, b_u, W_{xr}, W_{hr}, b_r, W_{xh}, W_{hh}, b_h)$. In our case, the x_t correspond to word embeddings: if s is a sentence of length L, then it is first converted into a list $(w_1, ..., w_L)$ of one-hot vectors, and each one-hot vector is mapped to a word embedding using a lookup matrix W_e. Therefore, we have $(x_1, ..., x_L) = (W_e w_1, ..., W_e w_L)$. The coefficients of W_e are learned, but they are initialized to precomputed word embeddings to avoid overfitting problems.

Capsules were designed by [20] for image processing. The idea behind capsules for computer vision consists in making complex computations and outputting a pose vector and an activation. This output is then routed towards subsequent capsules according to some predefined routing algorithm. The goal of that architecture is to have each capsule learning to recognize a visual feature based on what previous capsules have recognized before. For instance, some capsules could recognize eyes, a nose, a mouth and their respective positions. Then they would send their outputs to another capsule aiming at recognizing a whole face. It is architectured to avoid losing spatial information as common CNN do due to pooling operations. We think that capsules can also successfully perform other tasks such as NLP-related tasks, as our proposed model does.

In a nutshell, what we would like to do is to produce different embeddings that would attend to different semantic sides of the input sentence. A sentence would be divided into sub-sentences, and each of those sub-sentences would attend to a particular element of an image. These sub-sentences representations are then processed to build an embedding for the whole sentence.

In our model, all capsules share the same parameters and are similar to GRUs. In the following, we will explain the differences between them and actual GRUs. A recurrent capsule layer should process a sentence word-by-word and make updates in a way that would put attention on important words: the hidden state of each capsule should reflect one semantic side of the input sentence. Therefore, we need to define a routing procedure depending on current states and incoming words. For that purpose, we will use hidden states of capsules at time $t-1$ and the incoming word x_t to find how relevant a word is to a given capsule. More formally, if we consider the k-th capsule with $k \in \{1, ..., N_c\}$, update gates and reset gates will be the same as for a GRU:

$$u_t^{(k)} = \sigma(W_{xu}x_t + W_{hu}h_{t-1}^{(k)} + b_u), \tag{5}$$

$$r_t^{(k)} = \sigma(W_{xr}x_t + W_{hr}h_{t-1}^{(k)} + b_r), \tag{6}$$

We also compute $\tilde{h}_t^{(k)}$ as we do in a GRU:

$$\tilde{h}_t^{(k)} = \tanh(W_{xh}x_t + W_{hh}(r_t^{(k)} \circ h_{t-1}^{(k)}) + b_h), \tag{7}$$

We would like to make our routing procedure trainable via gradient descent, so we need to define differentiable operations. For that purpose, we will assume that for each capsule, for a given word w_t, we have a coefficient $p_t^{(k)} \in [0, 1]$ such that

$$h_t^{(k)} = (1 - p_t^{(k)})h_{t-1}^{(k)} + p_t^{(k)}\hat{h}_t^{(k)} \tag{8}$$

with

$$\hat{h}_t^{(k)} = u_t^{(k)} \circ \tilde{h}_t^{(k)} + (1 - u_t^{(k)}) \circ h_{t-1}^{(k)}, \tag{9}$$

which is the actual update computed in a GRU. The coefficient $p_t^{(k)}$ is a routing coefficient, describing to what extent a given capsule needs to be updated by the incoming word. As in [11], routing can be seen as an attention mechanism, putting attention on relevant words in our case. However, while the authors of [11] use Gaussians determined by EM-routing to compute this coefficient, we propose to compute it in a simpler manner. More details are provided in the next section. We can expand the last equation to get the following update:

$$h_t^{(k)} = (1 - p_t^{(k)}u_t^{(k)}) \circ h_{t-1}^{(k)} + p_t^{(k)}u_t^{(k)} \circ \tilde{h}_t^{(k)} \tag{10}$$

We can notice that it boils down to multiplying the update $u_t^{(k)}$ by a coefficient $p_t^{(k)}$. Then, how to compute $p_t^{(k)}$? For that purpose, we define an activation coefficient $a_t^{(k)}$ for each capsule:

$$a_t^{(k)} = |\alpha_k| + \log(P_t^{(k)}). \tag{11}$$

In the last equation, the α_k are random numbers drawn from a normal probability distribution (we found that 0.1 and 0.001 were good values for the mean and the standard deviation of the normal probability distribution). The α_k are important to our model because all capsules share the same parameters: if all activations are the same when they start processing a sentence, they will be all the same at the end. These random numbers break the symmetry between capsules; this is needed for our model to work properly. We assume $P_t^{(k)}$ ought to represent the semantic similarity between the current hidden state of the capsule $h_{t-1}^{(k)}$ and the incoming word x_t: if the incoming word is semantically similar to the previous hidden state, $P_t^{(k)}$ should be high, and if it is different, then it should be low. One can intuitively imagine that the cosine similarity $\cos(h_{t-1}^{(k)}, \hat{h}_t^{(k)}) = \frac{\left\langle h_{t-1}^{(k)} | \hat{h}_t^{(k)} \right\rangle}{\|h_{t-1}^{(k)}\|_2 \times \|\hat{h}_t^{(k)}\|_2}$ corresponds to a relevant definition of the semantic similarity between the current hidden state of the capsule and the incoming word: if the incoming word has a different meaning than previous words, then one can expect that $\hat{h}_t^{(k)}$ will reflect that different meaning. Therefore we define $P_t^{(k)}$ as:

$$P_t^{(k)} = \cos(h_{t-1}^{(k)}, \hat{h}_t^{(k)}). \tag{12}$$

Then we can compute $p_t^{(k)}$ according to the following formula:

$$p_t = \frac{\text{softmax}(\frac{a_t^{(1)}}{T}, ..., \frac{a_t^{(N)}}{T})}{M} \tag{13}$$

where M is the maximal coordinate of the vector $\text{softmax}(\frac{a_t^{(1)}}{T}, ..., \frac{a_t^{(N)}}{T})$ and T is a hyperparameter controlling the sharpness of the routing procedure (the higher T, the more we have one routing weight equal to 1 and all others equal to 0).

Our routing is different from those that were introduced in [20, 11]: the outputs of capsules are not combinations of all previous capsules outputs. Only the weights of the routing procedure depend on these previous capsules outputs.

Please note that if $T \to +\infty$, then all capsules receive the same inputs and produce the same hidden states: it is strictly equivalent to a GRU. Therefore, GRCs are an extension of the GRUs. The interest of GRCs over GRUs is that they can provide different representations of the same sentence, with attention put on some relevant parts of it. This idea is shown on Figure 3. Moreover, a GRC has the same number of trainable parameters as a GRU, but it has the ability to make more complex computations: for that reason we think that this architecture could be successfully used for other tasks than caption retrieval.

The model we propose for caption retrieval is similar to Word2VisualVec, but we replace the GRU by a GRC, as shown on Figure 1. Instead of concatenating the last hidden state of a GRU to a Word2Vec and a bag-of-words representations, we concatenate the average of the last hidden states of a GRC. We also tried to derive a weighted average of the hidden states of a GRC based on a soft-attention mechanism described in [5] but results did not improve. We reported our results in Section 4.3 for information.

3.3 Improving Word2VisualVec with GRC

As we said in Section 3.1, Word2VisualVec relies on three representations of sentences: bag-of-words, average of Word2Vec embeddings and GRU. GRCs provide another representation that we can concatenate to the three previous ones. More

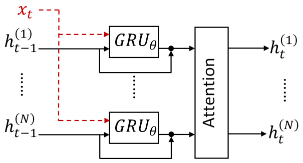


Fig. 3. Gated Recurrent Capsules: all capsules share the same learned parameters θ. The inputs of capsule i at time t are a word embedding x_t and its hidden state at time $t-1$ $h_{t-1}^{(i)}$. Its output is $h_t^{(i)}$, and it is computed through the routing procedure described in Section 3.2. This routing procedure can be seen as an attention model: each output depends on how semantically similar the incoming word is to previously processed words. It ensures that each capsule generates a sentence embedding corresponding to one important visual element of the sentence.

precisely, let us assume that we processed a sentence of length L with a GRC containing N_c capsules. Then, if $h_L^{(1)}, ..., h_L^{(N_c)}$ are the final hidden states of its capsules, the corresponding representation v_{GRC} of the sentence is the average of all these hidden states:

$$v_{GRC} = \frac{1}{N_c} \sum_{k=1}^{N_c} h_L^{(k)}. \tag{14}$$

This representation is intermediate between the GRU and the Word2Vec representations: it is the sum of N_c different hidden states, each of them corresponding to a particular part of a whole sentence.

Our goal is to map sentences to corresponding images in a space of visual features. One way to measure the efficiency of that kind of mappings is to evaluate the model on caption retrieval. When the model projects both images and sentences in a common multimodal space, recent works have shown that triplet ranking losses were efficient [7]. However in our case, sentences are directly mapped to a space of visual features, no transformation is made on image feature vectors. We found, in accordance with [6], that using the mean squared error (MSE) gave better results than a triplet ranking loss. Therefore, considering a mini-batch $B = ((s_1, x_1), ..., (s_{N_b}, x_{N_b}))$ of sentence-image pairs (N_b is the size of the mini-batch), we defined the loss function $\mathcal{L}_{MSE}(B)$ as follows:

$$\mathcal{L}_{MSE}(B) = \frac{1}{N_b} \sum_{k=1}^{N_b} \| f_\theta(s_k) - \phi(x_k) \|_2^2 , \tag{15}$$

where ϕ is a function mapping images to image features and f_θ is a function mapping sentences to image features where θ is the set of all trainable parameters. Our objective is to find a $\hat{\theta}$ minimizing $\mathcal{L}_{MSE}$:

$$\hat{\theta} = \text{argmin}_\theta(\mathcal{L}_{MSE}(\bar{B})) \tag{16}$$

where $\bar{B}$ is the set of all possible image-sentence pairs. We use the RMSProp method to optimize f_θ, following the procedure we describe in Section 4.2.

4 Comparison with Word2VisualVec

4.1 Dataset

We evaluated how our models performed on the caption retrieval task on the MSCOCO dataset [17]. This dataset contains 123000 images with 5 captions each, and we split it into a training set, a validation set and a test set according to [14]. The training set contains 113000 images, the validation set contains 5000 images and the test set contains 5000 images.

As for data preprocessing, we converted all sentences to lowercase and removed special characters (apart from spaces and hyphens). We limited the vocabulary to 5000 most used words, and replaced all other words by an "UNK" token. Regarding images, we projected them to a space of visual features. For that purpose, we used the penultimate layer of the ResNet-152 from [7] to get 2048-dimensional features vectors.

4.2 Parameters

Regarding the sentence embedding part of our model, we set its parameters as follows: we set the maximum sentence length to 24 (if longer the sentence is cut after the 24-th word). We initialized W_e using 500-dimensional Word2Vec embeddings trained on Flickr. We also used these embeddings to compute the Word2Vec part of sentences representations. These embeddings are the same as the ones that the authors of Word2VisualVec used in [6]. Regarding the GRC, we found that a model with 4 capsules and $T = 0.4$ performed well. The GRU in Word2VisualVec and the GRC capsules in our model have 1024-dimensional hidden states.

We trained our models using the RMSProp method [22] with mini-batches of 25 image-sentence pairs during 25 epochs. We followed the same learning rate decay procedure as in [6]: the learning rate was initially 0.0001 and we divided it by 2 when the performance of the model on the validation set did not increase during three consecutive epochs. We made all our implementations using the TensorFlow [1] library for Python and used the default parameters of the RMSProp optimizer: decay = 0.9, momentum = 0.0 and epsilon = 1e-10.

4.3 Results and Discussion

To prove the interest of our model, we compared it to Word2VisualVec. We compared the two versions we described in Section 3.2: the one with the average of final hidden states of capsules and the one with the soft-attention mechanism proposed in [5]. We reported our results in Table 1. They show that our model performs better than Word2VisualVec, and that the attention mechanism does not provide much improvement.

Moreover, we also wanted to see on which kind of sentences GRCs were more efficient than GRUs. For that purpose, we listed all the sentences that our model ranked in the top 9 sentences and that were ranked worse than rank

Table 1. Results of our experiments on MSCOCO. R@K denotes Recall at rank K (higher is better). Best results among all models are in bold.

MSCOCO			
Model	Caption Retrieval		
	R@1	**R@5**	**R@10**
Word2VisualVec	32.4	61.3	73.4
W2V + BoW + GRC	**33.4**	62.2	74.0
W2V + BoW + GRC + Attention	32.8	**62.3**	**74.2**

100 by Word2VisualVec. We also listed sentences ranked by Word2VisualVec in the top 9 that were ranked worse than rank 100 by our model. Our results are summarized in Table 2.

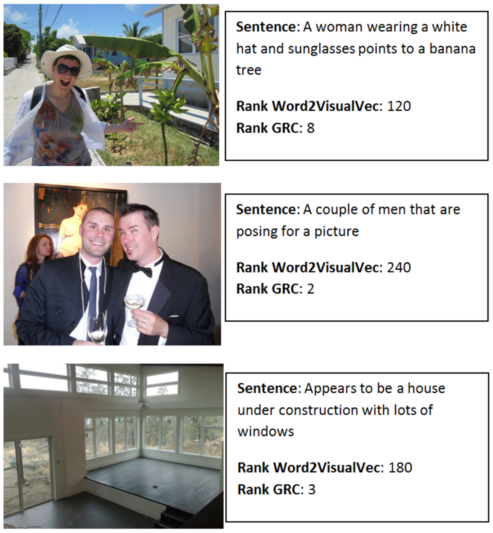
We noticed that sentences on which GRCs were outperforming GRUs were more likely sentences containing multiple visual concepts. We provide some examples in Figure 4. We think that this observation implies that GRCs could be used efficiently to derive finer visual sentence embeddings, taking into account important local elements. A possible direction of research would be to find how to combine it with an object detection model such as Faster R-CNN [19] to take advantage of that interesting property of GRCs.

5 Conclusion

In this paper, we introduced a novel RNN architecture called Gated Recurrent Capsules (GRCs). We built a model to address the caption retrieval task by mapping images and sentences to a visual features space. We showed in our experimental work that the models obtained using the proposed GRCs are surpassing those from earlier works (employing GRUs). Moreover, we stated that GRCs could potentially be used in any typical RNN tasks, as they are an extension of GRUs. An interesting future research direction would be to map outputs of capsules to local visual features.

Table 2. For each model: number of sentences ranked in top 9 for the right image by one model and above rank 100 by the other model. Ten sentences are ranked in top 9 by Word2VisualVec while ranked above rank 100 by our model, and seventeen sentences are ranked in top 9 by our model while ranked above rank 100 by Word2VisualVec. We also reported these numbers of sentences without counting sentences containing "UNK" tokens. This table shows that GRCs are performing better than GRUs on much more sentences than GRUs compared to GRCs.

	Word2VisualVec	**Our model**
Total	10	17
Total without UNK tokens	3	11



Sentence: A woman wearing a white hat and sunglasses points to a banana tree

Rank Word2VisualVec: 120
Rank GRC: 8

Sentence: A couple of men that are posing for a picture

Rank Word2VisualVec: 240
Rank GRC: 2

Sentence: Appears to be a house under construction with lots of windows

Rank Word2VisualVec: 180
Rank GRC: 3

Fig. 4. Compared results of Word2VisualVec and our model on three images.

Acknowledgments

One of the Titan Xp used for this research was donated by the NVIDIA Corporation. This work was partially funded by ANR (the French National Research Agency) via the GAFES project and the European H2020 research and innovation programme via the project MeMAD (GA780069).

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... and Kudlur, M. (2016, November). Tensorflow: a system for large-scale machine learning. In OSDI (Vol. 16, pp. 265-283).
2. Cho, K., van Merrinboer, B., Bahdanau, D., and Bengio, Y. (2014). On the Properties of Neural Machine Translation: EncoderDecoder Approaches. Syntax, Semantics and Structure in Statistical Translation, 103.
3. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). Ieee.
4. Dong, J., Li, X., and Snoek, C. G. (2016). Word2visualvec: Image and video to sentence matching by visual feature prediction. arXiv preprint arXiv:1604.06838.
5. Dong, J., Huang, S., Xu, D., and Tao, D. DL-61-86 at TRECVID 2017: Video-to-Text Description.
6. Dong, J., Li, X., and Snoek, C. G. (2018). Predicting visual features from text for image and video caption retrieval. IEEE Transactions on Multimedia.
7. Faghri, F., Fleet, D. J., Kiros, R., and Fidler, S. (2017). VSE++: improved visual-semantic embeddings. arXiv preprint arXiv:1707.05612.

8. Francis, D., Huet, B., and Merialdo, B. (2018, September). Embedding Images and Sentences in a Common Space with a Recurrent Capsule Network. In Proceedings of the 16th International Workshop on Content-Based Multimedia Indexing. IEEE.
9. Gu, Jiuxiang, et al. "Look, Imagine and Match: Improving Textual-Visual Cross-Modal Retrieval with Generative Models." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
10. 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 (pp. 770-778).
11. Hinton, G. E., Sabour, S., and Frosst, N. (2018). Matrix capsules with EM routing.
12. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
13. Karpathy, A., Joulin, A., and Fei-Fei, L. (2014, December). Deep fragment embeddings for bidirectional image sentence mapping. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2 (pp. 1889-1897). MIT Press.
14. 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 (pp. 3128-3137).
15. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
16. Lee, K. H., Chen, X., Hua, G., Hu, H., and He, X. (2018). Stacked Cross Attention for Image-Text Matching. arXiv preprint arXiv:1803.08024.
17. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... and Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
19. 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 (pp. 91-99).
20. Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In Advances in Neural Information Processing Systems (pp. 3856-3866).
21. Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).
22. Tieleman, T., and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.
23. Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).
24. Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320-3328).