Final Project: LEARN TO CAPTION

COMPSCI 689: Machine Learning

UMass Amherst

CICS

Fall 2016

Addanki, Raghavendra Kiran. raddanki@cs.umass.edu

Geffner, Tomas. tgeffner@cs.umass.edu

Shukla, Satya Narayan. snshukla@cs.umass.edu

Contents

1 Introduction

Generating captions for images represent an extremely challenging task; it does not only involve detecting correctly the main objects on an image, but also how are they interacting with each other, what are they doing, and where is the scene happening. A system able to generate correct captions for images automatically would be useful in several domains, such as early childhood education, helping visual impaired people navigate the web, and image retrieval, among others.

The task of captioning images can be naturally decomposed in several sub-tasks:
1) object detection;
2) determining what are the objects characteristics and how are these interacting; and
3) generation of grammatically correct sentences.

A lot of work has been done in each of these three areas, and good results have been achieved. For instance, regarding computer vision, in [4, 5] systems able to detect objects in static images with high accuracy were developed. Regarding Natural Language Processing (NLP), in [6] Rush et. al. describe a system able to summarize sentences, in [3] the authors describe a system able to translate texts from English to French, and in [1, 2] systems able to generate coherent and grammatically correct text were developed.

Methods used to generate image captions admit two possible classifications:

End to End vs. Pipeline: the former refers to methods that solve that solve each task separately (object detection, sentence generation) and combine the solutions, while the latter refers to methods that solve the tasks simultaneously.

Generative vs. Retrieval: the former refers to methods that generate the captions (word by word, character by character, etc.) and the latter to methods that pick the best caption available in a restricted set.

While the first methods developed fall under the pipeline category [7, 12], most recent work, which represents the state of the art, has been more focused on end to end models, such as those described in [11, 8, 14]. Most of these end to end models are based on deep convolutional neural networks and recurrent neural networks.

In this project, we build an end-to-end image caption generator, which given an image generates a reasonable description of the image in English.

2 Related work

The task of generating captions for images automatically has been approached in several different ways:

- In [7] the authors split the task into several sub-tasks and solve each one separately.
  1. Detectors are used to detect objects (bus, car, person, road, tree, ...)
  2. Each candidate object region is processed by a set of attribute classifiers
  3. Pairs of candidate regions are processed by preposition relational functions
  4. A Conditional Random Field (CRF) is generated using the previous information (see Fig. 2.1 for an example)

5. A labeling of the graph is predicted
6. The caption is generated

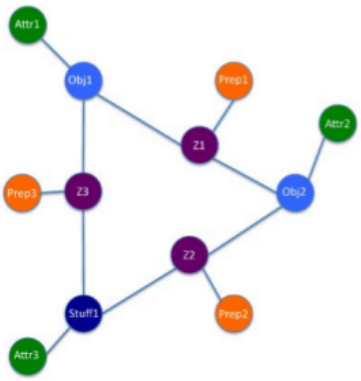


Figure 2.1: An example of a CRF with 3 objects (blue), 3 attributes (green) and 3 prepositions relating objects (orange/violet). Extracted from [7]

The output of the CRF is a predicted labeling which includes the objects, attributes and prepositions. Thus, it can be shown as triplets, as

$$<< blue, car >, next\ to, < big, tree >>$$

Based on this, the sentence must be generated. This task was formulated as an optimization problem with constraints. The objective function is the sentence's negative log-likelihood regarding the language model, while the constraints are given by the output of the CRF (the sentence has to describe the image).

- As mentioned in the introduction, captioning images can also be approached as a retrieving task.
  - In [12] the authors built a database of over 1 million captioned images. In order to do so, they first got a big and noisy set of images from Flickr using specific query terms, followed by a filtering of images in order to keep only those whose descriptions are relevant and visually descriptive. They managed this by keeping those images whose descriptions contained at least 2 of the terms used in the query, and one or more prepositional word (such as over, under, on, ...). Once the database is ready, the problem of generating a caption for a new image Y reduces to the problem of finding the image X in the database that maximizes $similarity(X, Y)$. This is done in two steps: 1) some images are filtered according to a global descriptor: 2) comparison of 4 types of features: objects and stuff, people, scene, TF-IDF weights. The images are ranked according to each content measure (of the 4 types of features) and then these results are combined in an overall ranking.
  - In [13] the authors describe a system to compute a score linking an image to a sentence. This can be used to generate caption for images, and to return an image that illustrates a sentence. This is done by mapping the image and the sentence to the same space, the *meaning space*, and comparing the *meaning* of the image to the *meaning* of the sentence. The representation used to express

meaning by the authors is a triplet of the form $< object, action, scene >$ (ie. $< bus, park, street >$). The problem of mapping images to the *meaning space* reduces to predict a triplet for the image, that the authors modelled as solving a multi-label Markov random field. The model consists on 3 nodes, one for the object, one for the action, and one for the scene. Each node can take a one value out of a discrete finite set (i.e. for the object node this set could be $\{dog, cat, bike, bus, man, woman, ...\}$). Using this the problem reduces to do inference on this model. In order to map the sentences to the *meaning space* a parser was used to generate a dependency parse, and the subject, direct object and some modifiers were extracted to compute the $< object, action >$ pair, and the scene information the head nouns of prepositional phrases were used. This two mappings are learned.

- In [8] Mao et. al. propose a deep learning to solve the image captioning problem, using a model they call multi-modal recurrent neural network. As it can be seen in Fig. 2.2, the m-rnn model contains a vision part, a language model part, and a multimodal part.

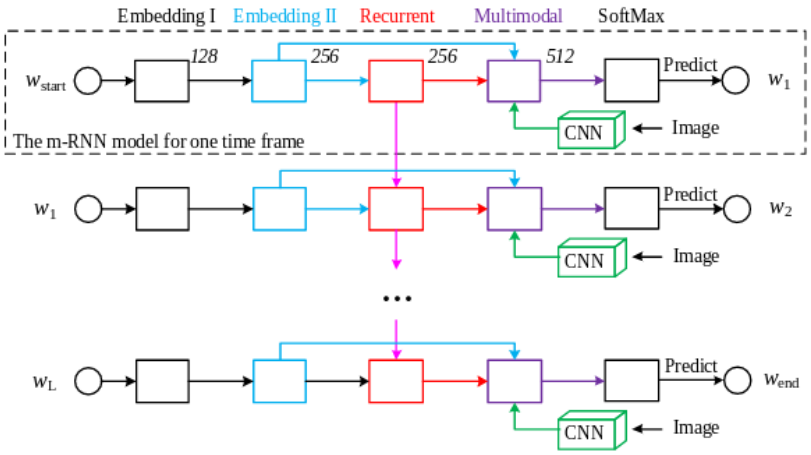


Figure 2.2: Image extracted from [8]

The language model consists on two densely connected layers used to get the embedding for each word, followed by a recurrent layer used to store temporal relations. The vision part consists on a convolutional neural network used to extract features from the images (they used the pre-trained AlexNet [9] or the VggNet [10]). Finally, the multimodal part connects the language model with the deep CNN. They tested their method on 4 datasets: IAPRTC-12, Flickr 8k, Flickr 30k, and MSCOCO.

3 Model

In this project, we use neural networks to generate image captions. Current developments in the area of machine translation show that sequences can be modeled accurately by directly maximizing the probability of correct translations given an input sentence. These approaches use recurrent neural network which takes a variable length sentence as input and converts it to a fixed dimensional vector representation in order to decode

it to a desired output sentence. We use the same principle here with one major change that the input here is an image (instead of a sentence) which has to be decoded into the natural language (caption).

The probability of the correct description given the image is maximized by using the following formulation:

$$\theta^* = \arg\max_{\theta} \sum_{I,S} \log p(S|I;\theta) \tag{3.1}$$

where θ are the parameters of our model, I is an image, and S its correct transcription. Since S represents any sentence, its length is unbounded. Thus, by applying the chain rule to model the joint probability over $S_0, ..., S_N$, where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, ..., S_{t-1}) \tag{3.2}$$

During training, the sum of the log probabilities is optimized as described in eq (3.2) over the whole training set using stochastic gradient descent.

A convolutional neural network is used to encode the image into a feature vector and then a language model is used to model $p(S_t|I, S_0, ..., S_{t-1})$, where the total number of previous words conditioned on at $t-1$ depends the length of hidden state or memory h_t. This memory is updated after seeing a new input x_t by using a non-linear function f:

$$h_{t+1} = f(h_t, x_t) \tag{3.3}$$

For f we use a Long-Short Term Memory (LSTM) unit and Recurrent Neural Network (RNN), which has shown state-of-the art performance on sequence tasks such as translation. They have been described in detail in following sections.

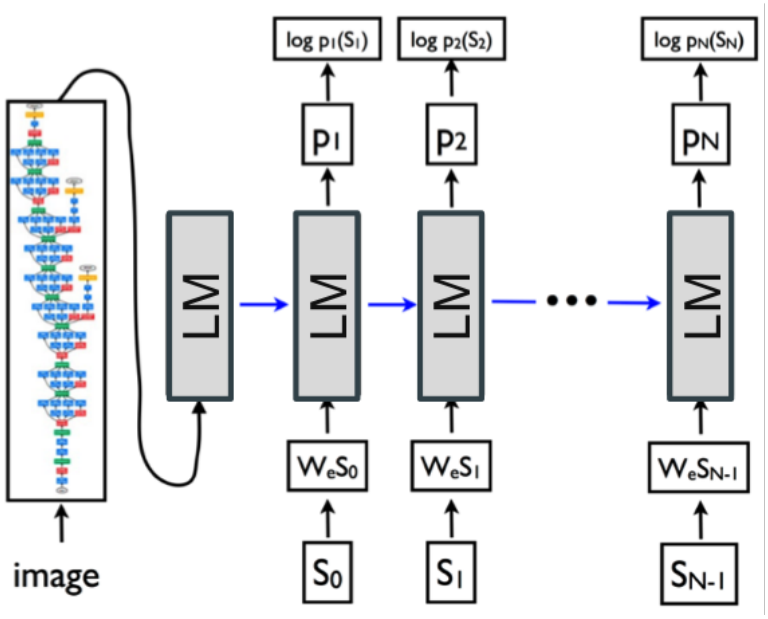


Figure 3.1: Language model (LM) combined with a CNN image embedder and word embeddings. All LMs share the same parameters.

3.1 CNN

We use a Deep Convolutional Neural Network to extract the image's features. This CNN is a pre-trained model, which is not trained with the complete system (it can be trained, but for computational reasons we decided not to). This particular network uses batch normalization [17] and yields one of the best performance on ImageNet Large Scale Visual Recognition Competition 2014. A detailed description of the network used can be found in [16]. As it can be seen in Fig. 3.1, the image is fed only once to the system, which is able to retain the information during the task thanks to the Language Model. After, the sentence is generated word by word, using as inputs the last word generated until the "end symbol" is generated.

3.2 Recurrent Neural Networks

Recurrent Neural Networks have shown great promise in many natural language processing tasks. The core idea behind RNNs is to make use of sequential information. In traditional neural networks, we generally assume that all input (and corresponding outputs) are independent of each other but it is not true in many cases. Suppose, if we are trying to predict the next word of a sentence, we would want to know the whole context/words that came before it. RNN's are called recurrent because they perform same task again and again for every element in the sequence with the output being dependent on previous computations. They can be thought of as "memory" blocks which stores the information about previous computations. In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps.

$$s_t = f(Ux_t + Ws_{t-1}) \tag{3.4}$$

$$o_t = \text{Softmax}(Vs_t) \tag{3.5}$$

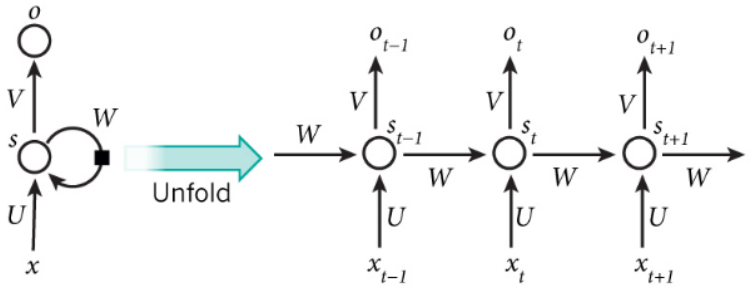


Figure 3.2: A recurrent neural network and the unfolding in time of the computation involved in its forward computation. Source: Nature

RNN shares the same parameters (U, V, W above) across all steps. This reflects the fact that we are performing the same task at each step, just with different inputs. This greatly reduces the total number of parameters we need to learn. The main feature of an RNN is its hidden state, which captures some information about a sequence.

3.3 Long Short Term Memory Networks (LSTM)

The choice of f in eq (3.3) is governed by its ability to deal with vanishing and exploding gradients [20], the most common challenge in designing and training RNNs. To address this challenge, a particular form of recurrent nets, called LSTM, was introduced [20] and applied with great success to translation [3], [21] and sequence generation [2].

The core of the LSTM model is a memory cell c encoding knowledge at every time step of what inputs have been observed up to this step (Fig. 3.3) . The behavior of the cell is controlled by "gates" - layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zero this value if the gate is 0. In particular, three gates are being used which control whether to forget the current cell value (forget gate f), if it should read its input (input gate i) and whether to output the new cell value (output gate o). The definition of the gates and cell update and output are as follows ($**$ represent the component-wise multiplication for matrices):

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \tag{3.6}$$
$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \tag{3.7}$$
$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \tag{3.8}$$
$$c_t = f_t * * c_{t-1} + t_t * * h(W_{cx}x_t + W_{cm}m_{t-1}) \tag{3.9}$$
$$m_t = o_t * * c_t \tag{3.10}$$

where x_t represents the input to the LSTM layer, m_{t-1} the output for the previous timestep, and the matrices W_{ab} are parameters that are learned. Finally, the output after the softmax layer is given by

$$p_{t+1} = \text{Softmax}(m_t)$$

Such multiplicative gates make it possible to train the LSTM robustly as these gates deal well with exploding and vanishing gradients [20]. The nonlinearities are sigmoid $\sigma(\cdot)$ and hyperbolic tangent $h(\cdot)$.

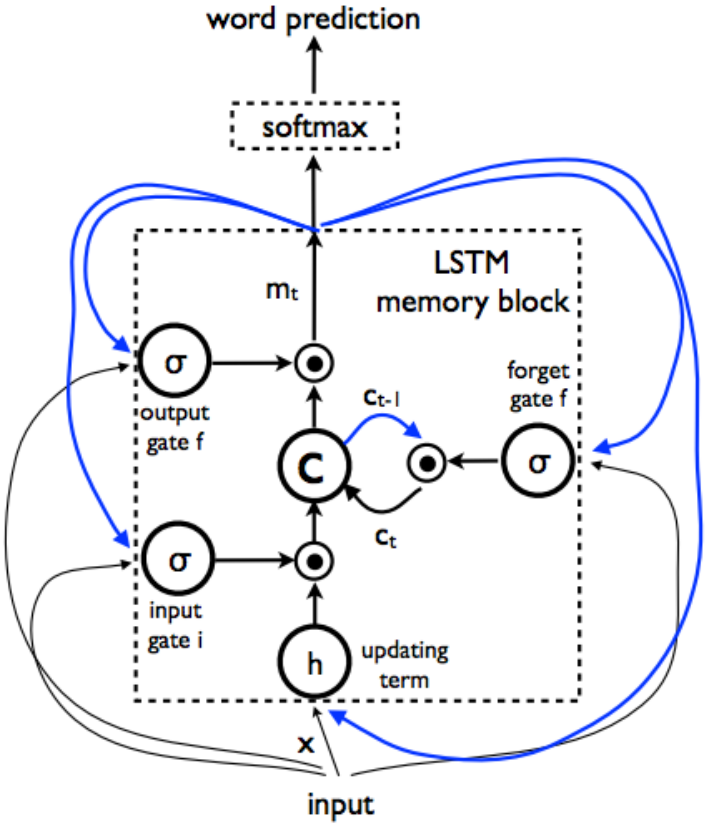


Figure 3.3: LSTM: the memory block contains a cell c which is controlled by three gates. In blue we show the recurrent connections - the output m at time $t-1$ is fed back to the memory at time t via the three gates; the cell value is fed back via the forget gate; the predicted word at time $t-1$ is fed back in addition to the memory output m at time t into the Softmax for word prediction.[11]

3.4 Training

The Language model (LM) 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 $p(S_t|I, S_0, ..., S_{t-1})$. We can think of this as a copy of the memory in language model is created for the image and each sentence word. Output m_{t-1} of the LM at time $t-1$ is fed to the LM at time t (see Fig. 3.1). All recurrent connections are transformed to feed-forward connections in the unrolled version. In more detail, if we denote by I the input image and by $S = S_0, ..., S_N$, a true sentence describing this image, the unrolling procedure reads:

$$x_{-1} = \text{CNN}(I) \tag{3.11}$$

$$x_t = W_e S_t, t \in 0...N-1 \tag{3.12}$$

$$p_{t+1} = \text{LM}(x_t), t \in 0...N-1 \tag{3.13}$$

where each word is represented as a one-hot vector S_t of dimension equal to the size of the dictionary. Note that S_0 denotes a special start word and by S_N a special stop word which designates the start and end of the sentence. In particular by emitting the stop word the LM signals that a complete sentence has been generated. Both the image and the words are mapped to the same space, the image by using a vision CNN, the words by using word embedding W_e. The image I is only input once, at $t = -1$, to inform the LM about the image contents.

Our loss is the sum of the negative log likelihood of the correct word at each step as follows:

$$L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t) \tag{3.14}$$

The above loss is minimized w.r.t. all the parameters of the language model, the top layer of the image embedder CNN and word embeddings W_e.

3.5 Inference

There are several approaches to generate a sentence from an image. The first one is Sampling where we just sample the first word according to p_1, then provide the corresponding embedding as input and sample p_2, continuing like this until we sample the special end-of-sentence token or some maximum length. The second one is Beam Search: 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. We used Beam search approach in the experiments here.

4 Experiments

We performed an extensive set of experiments on our dataset varying the type of units, number of layers in the language model (shown in Fig. 3.1) to assess the performance of this method against state of the art methods and various non-deep learning approaches used in earlier works.

4.1 Dataset

For training and evaluation, we use MSCOCO (Microsoft Common Objects in Context) dataset[22]. Containing photos of 91 objects types that would be easily recognizable by a 4 year old, MSCOCO is a very popular dataset on which most of the image captioning work is based on. It has about 120,000 images with 80,000 images used during training and rest of them used for validation. Each of these images is also provided with at least 5 human annotated captions describing it.

Figure 4.1: MSCOCO dataset: Image and annotated captions describing it.

	Training	Validation	Testing
Total Images	82784	40505	1500

Table 4.1: MSCOCO Image Captioning Dataset.

Of these 40,505 validation set of images, we use 1500 images for testing (these were not used in any way during training).

4.2 Evaluation Metrics

The natural question of how an image captioning system can be evaluated arises. Ideally, human evaluators check the generated captions and give them scores depending on the grammatical correctness and on whether it describes the image correctly or not. Clearly, this is not efficient, nor objective. When developing these systems is normal to try several values for the model hyper parameters, and different modification in the main architecture, so testing the system and checking whether an improvement has been achieved should be fast. Should humans evaluators check and score the captions, it could take days or even weeks to determine the performance of a system on a test set of thousands of images. In response to this, in [15], Papineni et. al. developed an automatic method to assign scores to a generated text given a reference. This score is called BLEU (Bilingual Evaluation Understudy). Currently this method is used as one of the references to evaluate caption generating systems.

Basically, to compute the BLEU-n score of a generated sentence the quantity $S = com/cand$ is computed, with com being the number of common n-grams between the generated sentence and the reference, and $cand$ being the number of n-grams in the generated sentence. The value of S is then modified taking into account the differences in length between the original and generated sentence, and the maximum number of times that an n-gram appears in the reference. Notice that the BLEU score is always between 0 and 1. As we will see in the results, many systems obtain higher BLEU scores than human generated captions, which might indicate that despite being useful and efficient, BLEU scores have some drawbacks.

4.3 Training Details

The task of assigning a description is strictly harder than object classification and data driven approaches have only recently become dominant thanks to datasets as large as ImageNet and MSCOCO. Our model is an example of an encoder-decoder neural network. It works by first encoding an image into a fixed-length vector representation, and then decoding the representation into a natural language description. The image encoder is a deep convolutional neural network.

Our particular choice of encoder network is the Inception v3[16] image recognition model pretrained on the ILSVRC-2012-CLS (ImageNet Large Scale Visual Recognition Challenge 2012) image classification dataset. Inception-v3 is trained for the ImageNet Large Visual Recognition Challenge 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes, like Zebra, Dalmatian, Dishwasher etc. This model is currently the state-of-the-art on ImageNet dataset for classification.

The decoder is a commonly used for sequence modeling tasks such as language modeling and machine translation. We use the following different models for the decoder network varying the type of units and the number of layers in the network.

- 1 LSTM layer
- 2 LSTM layers
- Typical RNN cells instead of LSTM (one layer)

Given an image, our encoder extracts the features of the image - a 512 length vector, using a deep CNN with pretrained weights obtained from Inception v3 model. This vector is then input into the language model which can be of various types as described above. A start symbol is then input into the language model to initiate the process of sentence generation. The network then outputs a set of symbols based on the probability of occurrence. This set of words or symbols is then fed into the language model and this process continues until the maximum sentence length is reached or the language model outputs end symbol. Each of these symbols is represented as a 512 length vector and the embedding of these words in the vector space is also jointly learnt during training.

Using different models for the decoder network - 1 LSTM layer, 2 LSTM layers and 1 layer RNN, we trained all sets of weights using stochastic gradient descent with fixed learning rate and no momentum. All weights were randomly initialized except for the CNN weights in the encoder network. Our models were trained on a CPU with 16GB RAM and an Intel i7 processor. We trained for about 7000 iterations with a batch size of 32 which took us about 20-25 hours per model.

5 Results

In order to compare our results with non-deep learning approaches, we used the following methods.

- We used a retrieval based approach in this method [12]. From a database of over 1 million captioned images, we obtain the image closest to the test image using a set of criteria such as objects, people, scene etc. and output the caption of the closest image.
- We used a nearest neighbour approach(kNN) in this method. It is similar to the previous method except that the similarity function is different than the previous one. Here, we look for an image closest to the test image in the feature space.
- Human Agreement - Human scores were computed by comparing one of the human captions against the other four. We do this for each of the five captions, and average their BLEU scores.

Apart from this, we test it against a state-of-the art image captioning system Neural Talk 2[14]. This model is trained for approximately 1 million iterations.

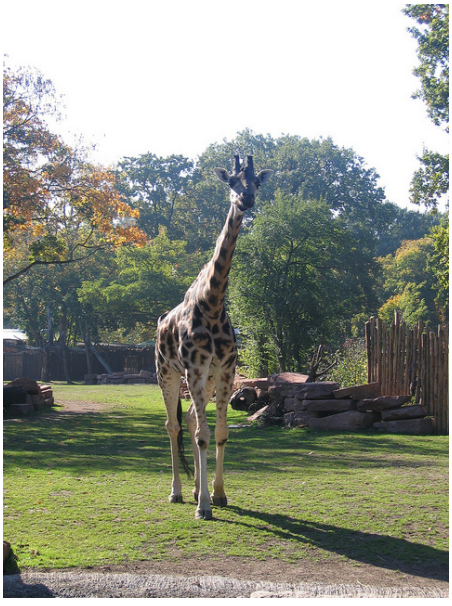
	BLEU 1	BLEU 2	BLEU 3	BLEU 4
CNN + 1 layer LSTM	0.9	0.79	0.69	0.61
CNN + 2 layer LSTM	0.89	0.79	0.69	0.61
CNN + 1 layer RNN	0.92	0.73	0.52	0.35
Neural Talk 2	0.90	0.84	0.77	0.71
Retrieval Task	0.69	0.53	0.35	0.25
Nearest Neighbour	0.48	0.28	0.17	0.10
Human Agreement	0.62	0.45	0.30	0.22

Table 5.1: BLEU scores obtained by different systems

Of all the BLEU-n scores listed, BLEU 4 is considered to be a good indicator for sentence similarity. From the Table 5.1, it is evident from BLEU 4 scores, that the methods using neural networks outperform rest of the approaches with a significant margin. We observe that Human Agreement on BLEU-n scores gives very low scores when actually the scores should be close to 1. This suggests that BLEU-n is not such a good metric and doesn't take into account the synonyms , language structure and finer aspects of a language. Also, another observation to be noted is that **CNN + 1 layer RNN** model performs rather poorly in comparison to LSTM based models. This might be due to the fact that RNN might not be able to retain the past information in a sentence as effectively as LSTM which is generally expected to be true.

Our system achieves best performance using **CNN + 1 layer LSTM** model. It is noteworthy that even with about 7000 iterations we were able to achieve similar BLEU scores as compared to Neural Talk 2 which was trained on about 1 million iterations.

We also provide captions generated by our system for a few images.



(a) A giraffe standing next to a fence in a field.

(b) A man riding a wave on a surf board.

(a) A woman holding a tennis racquet on a tennis court.

(b) A man is skiing down a snow covered slope.

(a) A herd of elephants standing next to each other.

(b) A double decker bus driving down a street.

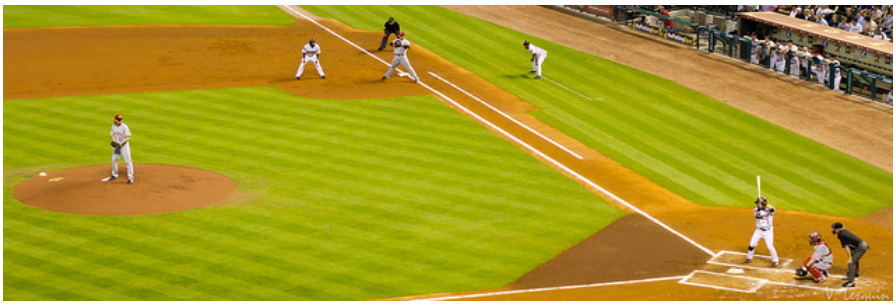


Figure 5.4: A baseball player swinging a bat on a court.

(a) A man sitting at a table with a cake.

(b) A man in a suit and tie standing in front of a mirror.

6 Conclusions

We have described an end-to-end neural network system that can automatically view an image and generate a reasonable caption in English. This system is based on a convolu-

tion neural network that encodes an image into a compact representation, followed by a recurrent neural network(LSTM) that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. Experiments on MSCOCO dataset show the robustness of our system in terms of qualitative results (the generated sentences are very reasonable) and quantitative evaluations, using BLEU, a metric used in machine translation to evaluate the quality of generated sentences.

With more image data being available, it seems that this problem will become a central problem in vision and machine learning research. Further extensions or research directions could involve better evaluation metrics because of the drawbacks of BLEU scores (noted in section 5), obtaining a language independent model for image captioning; recent works of Google Language Translation suggests that this might be possible. Application centric captions, searching on the images - querying etc are also interesting research directions involving the problem of image captioning.

References

[1] Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating text with recurrent neural networks." Proceedings of the 28th International Conference on Machine Learning (ICML). 2011.

[2] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850 (2013).

[3] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[4] Kim, Kye-Hyeon, et al. "PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection." arXiv preprint arXiv:1608.08021 (2016).

[5] Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).

[6] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[7] Kulkarni, Girish, et al. "Babytalk: Understanding and generating simple image descriptions." IEEE Transactions on Pattern Analysis and Machine Intelligence 35.12 (2013): 2891-2903.

[8] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." arXiv preprint arXiv:1412.6632 (2014).

[9] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[10] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[11] Vinyals, Oriol, et al. "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge." IEEE Transactions on Pattern Analysis and Machine Intelligence (2016).

[12] Ordonez, Vicente, Girish Kulkarni, and Tamara L. Berg. "Im2text: Describing images using 1 million captioned photographs." Advances in Neural Information Processing Systems. 2011.

[13] Farhadi, Ali, et al. "Every picture tells a story: Generating sentences from images." European Conference on Computer Vision. Springer Berlin Heidelberg, 2010.

[14] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[15] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

[16] Szegedy, Christian, et al. “Rethinking the inception architecture for computer vision.” arXiv preprint arXiv:1512.00567 (2015).

[17] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in ICML, 2015.

[18] 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,” in ICML, 2014.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in ICLR, 2013.

[20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, 1997.

[21] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in EMNLP, 2014.

[22] Lin, Tsung-Yi, et al.“Microsoft COCO: Common objects in context.” European Conference on Computer Vision. Springer International Publishing, 2014.