Review of deep learning approaches for image caption generation

Ayushman Singh Sisodiya

May 27, 2016

1 Introduction

Generation of description of images using natural language sentences is gaining much popularity these days. It is a challenging task, as it requires not only understanding an image, but to translate that visual knowledge into sentence description. To capture the correlation between two modalities i.e. visual and natural language we need to map both these to some same space so as to learn the relation between them or say we need to learn the multimodal joint model. The traditional way to do this was to use sentence template or do a retrieval task based on ranking. These methods doesn’t work or generalize well to new images. We somehow need to map both of them to a common semantic space. Again deep neural networks come to the rescue here due to their power to form new grammatically correct sentences as opposed to the template based models and their generalization capability to a novel image.

2 Datasets

The most popular datasets used for this task and their brief description is given below

2.1 Flickr8K

It has 8000 images and has 5 captions for each image. The standard dataset division that are used for this set is 6000 for training, 1000 for validation, 1000 for testing.

2.2 Flickr30K

It is an extension of the above Flicker8k dataset. It has 31783 images with 5 full sentence level caption for each image. The public available dataset provided by Karpathy, has the following division of the dataset. 28000 for training, 1000 for validation and 1000 for testing.

2.3 MSCOCO

This is the most recent and probably more used and standardized dataset. It has 82783 images for training and 40504 for validation and testing. Each image has 5 captions. Since the provided split of dataset is too skewed, the followed division is followed 80000 for training and 5000 both for validation and testing.

2.4 Visual Genome Dataset

This dataset consists of 94000 image and 4100000 captions for different regions in the images. All the captions are human generated on Amazon Mechanical Turk and the descriptions are provided with a bounding box with the corresponding region. This image set is taken from the intersection of MSCOCO and YFCC100M datasets. Each image has an average of 21 objects, 18 attributes and 18 pairwise relationships between objects. This relationships is provided by the metadata of the dataset.
2.5 **SBU**

This dataset contains 1 millions images and their description collected from the flickr website by performing huge number of queries and then removing the noisy data. This dataset is generally not preferred cause it relies on the description given by the image owner. So this dataset is said to be biased and has more noise.

2.6 **Pascal Voc Dataset**

This dataset is build around the pascal voc 2008 dataset which is a dataset for object classification task. They take 50 images from each of the 20 classes and use the amazon's mechanical turk to generate 5 captions for each image. They also label the image with a triplet which consists of the mail object in the image, the main action and the main place.

2.7 **PASCAL-50S**

This dataset is based on the popular UIUC pascal sentence dataset. It has 50 human generated description for each image and has 1000 such images.

2.8 **ABSTRACT-50S**

This dataset is based on dataset by Zitnick and Parikh. It also has 50 human generated description for each image and has 254721 such images.

3 **How its done?**

Image caption generation is a core part of scene understanding which is one of the primary goals of vision right from the start. Over the years many different approaches has been developed. We can broadly classify them into three categories, which are as follows

1. **Template based** - The main idea here is to detect the objects and their attributes, parse the sentence into phrases and learn their correspondence using models like conditional random fields(CRFs). These methods generalize poorly because of their limitation to generate variable length sentences

2. **Retrieval based** - These leverage the distance in the visual space to retrieve images which are similar to the test image and then modify and combine their captions to form the caption for the test image. These are not the model we desire as they highly depend on the training or seen data and need additional steps like modification and generalization to output the final caption.

3. **Neural Network based** - These are the models where we will focus the most. These are the models that uses different deep neural networks like convolutional neural network(CNN), long short term memory(LSTM) networks, recurrent neural network(RNN) to implicitly learn the common embedding by encoding and decoding the different modalities. These by far gives the best result on all common datasets of caption generation.

We will be paying our attention to the third category and will discuss the recent advances in it below

3.1 **Show and Tell: A Neural Image Caption Generator**

The link to the paper is [here](#).

3.1.1 **Intro**

- This is the most basic model and probably the first model that was end to end trainable.
- [Vinyals et al., 2015](#) try to maximize the log likelihood of the probability distribution of sentence conditioned on the image. The equation shows the mathematical formulation

\[
\theta^* = \arg \max_{\theta} \sum_{(I,S)} \log p(S|I; \theta)
\]  \(1\)
The inspiration came from the recent advances in machine translation, which instead of translating word to word and then reordering uses RNN, which was quite simple and still reaches state of the art performance.

For machine translation, a encoder RNN is used which produces a rich fixed length vector for the input sentence. Then this vector is fed to decoder RNN as the first hidden state with start marker as the input.

Vinyals et al. [2015] replaced this encoder RNN with CNN features of the image as the CNN features are widely used in all computer vision tasks. They called this model as Neural Image Caption (NIC).

They also show that feeding image with every word at training and testing time decreases the quality of the results.

The use of beam search at the inference time, in which at any time $t$ you keep the best $k$ sentences until that time and use these $k$ sentences to get the sentence of length $t+1$ and then chose the best $k$ among them improved the result.

They used the BLEU evaluation metrics to evaluate and compare their result and perplexity of the model for tuning hyper parameters and model selection.

Perplexity is geometric mean of the inverse probability for each predicted word.

### 3.2 Deep Visual-Semantic Alignments for Generating Image Descriptions

The link to the paper is [here](#).

#### 3.2.1 Intro

- The datasets present for the task of image caption generation have huge number of sentences but it is not mentioned which part of the sentence correspond to which scene or object of the image.

- Karpathy and Fei-Fei [2015] wanted to learn these alignments and use this to generate the image description. The inference of alignment was done by embedding both image and sentence to a common space and the approach was tested on image-sentence retrieval task.

- They also propose a multimodal recurrent neural network model which generates captions for image and train this model on the inferred dataset and test this model on region level annotation based dataset created by them.

#### 3.2.2 Model

- To get the latent alignment of the sentence parts to their corresponding scenes in the image Karpathy and Fei-Fei [2015] build on a approach where dependency tree relations are grounded into the image regions with a ranking objective. Instead of this for the sentences part they used a bidirectional RNN to compute the word representation in the sentence. Now using neural network they map these word representations and image regions onto common space and train it in such a way that semantically similar concepts occupy nearby regions in this space.

- RCNN is used to get the top 19 regions based on the score in the original image and then these regions along with the original image is projected on the common space.

- Now as both the image and sentence words into a common space. A score of similarity is defined between them. It is intuitive to say that a word corresponds to a region if word support the region. So this can be seen as the dot product between the two. Therefore $v_i^T * s_t$ is the score between the $i^{th}$ region and $t^{th}$ word. And score between the whole image and full sentence can be seen as the sum of these individual region-word scores.

$$S_{kl} = \sum_{t \in g_k} \sum_{i \in g_k} \max(0, v_i^T * s_t)$$  \hspace{1cm} (2)
This interpretation of the score is that sentence fragments align to a subset of image regions whenever the dot product is positive. This score is reformulated so as to simplify the model and reduce the additional objective and hyper parameters.

\[ S_{kl} = \sum_{t \in g_k} \max(v_t^i \ast s_t) \]  

(3)

- The simplified score tells that each word is associated with only one region in the image. The structured max margin loss is given below

\[ C(\theta) = \sum_k \left[ \sum_l \max(0, S_{kl} - S_{kk} + 1) + \sum_l \max(0, S_{lk} - S_{kk} + 1) \right] \]  

(4)

- \( k = l \) corresponds to the original image-sentence pair.
- The loss function forces the score of aligned image sentence pair to have high scores than that of misaligned pairs.
- The above formulation assigns each word with a region in image, which scatters the word of the sentence all over. Also to get contiguous phrases to be assigned to a region the authors model this problem as MRF with the true alignment of phrases as the latent variables in the model. The energy of the model is as follows

\[ E(a) = \sum_{j=1}^{N} \psi_U^j(a_j) + \sum_{j=1}^{N} \psi_B^j(a_j, a_{j+1}) \]  

(5)

\[ \psi_U^j(a_j = t) = v_t^i \ast s_t \]  

(6)

\[ \psi_B^j(a_j, a_{j+1}) = \beta [a_j = a_{j+1}] \]  

(7)

- This forces binary interaction between neighboring words to align in the same region
- \( \beta \) controls the size of phrases. Smaller the \( \beta \) smaller the phrases.
- Minimization of the energy and then using dynamic programming to get the image regions with sentence description phrase from the whole sentence.
- Now the multimodal RNN learning is done in a same way as in the Vinyals et al. 2015

3.3 DenseCap: Fully Convolutional Localization Networks for Dense Captioning

The link to the paper is here

3.3.1 Intro
- Johnson et al. 2015 propose a dense captioning task in which each region of an image is detected and a set of descriptions is generated for each region. Hence object detection task is a special case when descriptions are one word and image captioning when detected region is the whole image.
- Their main contribution is the introduction of the dense localization layer. This layer finds out regions of interest in the image and extracts the activations inside the region using a bilinear interpolation.

3.3.2 Model
- The authors used the VGG-16 conv-net for the image part of the dense caption generation and removed the final pooling layer from it. So given an image they get features of 512 uniformly random sampled regions which will now go to the localization layer.
- This localization layer takes the tensor and extract the ROI from the identified regions
- They used the technique from Faster RCNN, but replaced their final layer with bilinear interpolation and this gives the ability to predict morphed and affine regions.
• Now these features for regions are passed through a fully connected layer with ReLu units and regularized dropouts. This converts each region into a 4096 dimensional vector. A final adjustment is done to this box and score and offsets calculated are again calculated.

• Then a standard LSTM network is used for training with conditioning on the image feature vector. The loss function used is average cross entropy.

• Binary logistic loss is used for positive and negative regions and L1 loss for box regression. All losses were normalized according to the batch size and RNN sequence length.

• They used METEOR evaluation metrics since it was the most highly correlated with human judgment

3.4  Image Captioning with Deep Bidirectional LSTMs

The link to the paper is [here](#).

3.4.1  Intro

• [Wang et al. 2016](#) proposes two novel deep bidirectional variant models, in which they increase the depth of non-linearity transition in different ways and are said to learn hierarchical visual-language embedding

• The model performs comparatively with the state of the art even without additional mechanism like attention model or object detection

• Data augmentation techniques such as multi-crop, multi-scale and vertical mirror are used to prevent overfitting in training deep models

• They have used CNN to learn visual features alexnet and 16-layer vggnet to be precise and Bidirectional LSTM to learn sentence features

• The inspiration to use deeper Bi-LSTM comes from the human brain and the advances in the deep CNN

• End-to-end training is done by minimizing the joint loss

• LSTMs are already deep, but that depth is horizontal, in which weights are reused which limits the learning of more representative features. Vertical depth can be achieved by stacking multiple LSTMs or use a MLP in between which helps in making the net deeper with increasing too much parameters

• Their model is end to end trainable with stochastic gradient descent with BPTT algorithm. The loss at the end is the sum of loss from both forward and backward LSTMs.

• To infer the final sentence for the image they take that word which has the maximum probability from both forward and backward prediction.

• For word vector they used one-hot representation along with tokenizing and removing less occurring words.

3.4.2  Bi-multimodal-LSTM

• The authors propose a different variant of bidirectional LSTM.

• First there are two T-LSTM which learn the encoding of text input in both forward and backward direction.

• Then two multimodal LSTM which takes the corresponding forward and backward output of the T-LSTM along with the image features through CNN with each LSTM node to capture the strong correspondence between visual and semantic spaces

• Then there is a final summation just below the softmax layer which gives the probability distribution over words.
3.4.3 Deeper LSTM

- So instead of stacking the T-LSTM they proposed to use multilayer perceptron in between the LSTM transition. It has two fold profit.

- First, it is easier to train and second it highly reduces the number of parameters

3.5 Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

The link to the paper is here.

3.5.1 Intro

- Attention is important as representing the whole image from one vector is not the better way to go about scene understanding, which is what it is done in most cases.

- By using attention features can be dynamically used when needed which is really helpful when there is a lot of objects in the image.

- Until now features from the last layer from the conv-net is used which focus on the most important part of the image, which is one way to go but is not helpful when more descriptive captions are required.

- So more low level features are needed from the image, but to handle so much more information a mechanism is desired

- Two types of attention are introduced in the paper by Xu et al. [2015], hard attention and soft attention. The use of attention provides with the ability to visualize the model better.

3.5.2 Model

- Both the models i.e. soft and hard, have a common framework and differ only in calculation annotation vector weights, which we will describe later

- So as mentioned earlier, instead of using the final fully connected layer of CNN, the lower convolutional layer vectors are used called them annotation vectors. Each annotation vector is supposed to belong to some specific part of the image (which makes sense as this is the basic nature of the CNN)

- For the decoder part a slight variation of the widely used normal LSTM is proposed, in which along with the embedding of the word and the vector of the previous hidden state, a context vector is also passed which is to capture the visual information associated with a particular input location and is based on the previous generated word and the previous hidden state

- The calculation of this context vector is what is different in hard and soft visual attention and is the dynamic representation of the relevant part of input image at that time step.

- So at each time step, weights $\alpha_i$ are calculated for each of the annotation vectors of the locations and based on these $\alpha$’s and the annotation vectors $a_i$, $\phi$ selects one of the vectors as the context vector $z_i$ for that time step

- Also the weights $\alpha_i$ is calculated by a MLP with the previous hidden state $h_{t-1}$ as the input and softmax at the end to make the summation equal to 1

- The difference between the stochastic hard and deterministic soft attention is that they introduce latent parameter $s_{t,i}$ which indicates the location and are drawn from multinoulli parametrized by the $\alpha$’s while in the soft attention case they make it deterministic by taking the expectation instead of sampling which makes it differentiable and end to end trainable
4 Evaluation Metrics

4.1 BLUE-N: Bilingual Evaluation Understudy

\[ B_N = \min(1, e^{1-r/c} e^{(1/n) \sum_{n=1}^{N} \log p_n}) \]  

where \( r, c \) are the length of reference sentence and generated sentence respectively and \( p_n \) is the modified n-gram precisions.

4.2 ROUGE-N: Recall-Oriented Understand for Gisting Evaluation

Formally ROUGE-N is the recall of n-gram between the candidate and reference sentences. It is given by

\[ \text{ROUGE-N} = \frac{\sum_{S \in \text{ReferenceSentences}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ReferenceSentences}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \]

where \( n \) is the length of n-gram, \( \text{gram}_n \) and \( \text{Count}_{\text{match}}(\text{gram}_n) \) is the maximum number of n-gramms co-occurring in a candidate and set of reference sentences.

4.3 Meteor: Metric for Evaluation of Translation with Explicit ORdering

4.4 Intro

- This metric is derived from the harmonic mean of unigram precision and recall and is proposed by Banerjee and Lavie [2005]
- Recall is given more importance than the precision. So its weight is more.
- It basically has three levels of matching, exact word, stemmed and synonymy.
- It is said to have better correlation with the human judgement.

4.4.1 Algorithm

- The alignment is created between the candidate sentence and reference sentence, which is a mapping between the unigrams with the constraint that every unigram in the candidate string can map to 0 or 1 unigram in the reference string.
- The best alignment is chosen, which is the alignment with the most mapped unigrams and least crossed lines. Least crossed means the 1st unigram will be preferred to mapped to the 1st unigram rather than last (considering all three are same).
- Now unigram precision is calculated as follows

\[ P = \frac{m}{w_t} \]

where \( m \) is the number of unigrams in candidate sentence that are also found in the reference sentence and \( w_t \) is the number of unigrams in the candidate sentence.

- Unigram recall is calculated as follows

\[ R = \frac{m}{w_r} \]

where \( m \) is the same as above and \( w_r \) is the number of unigrams in the reference sentence.

- The above two score are combined as follows with recall given 9 times more weight than precision.

\[ F_{\text{mean}} = \frac{10PR}{R+9P} \]

- To capture more linguistics and semantics we need to do the alignment for n-grams and compute a penalty \( p \) for the alignment. The more the mappings are not adjacent in the candidate and reference sentence the more the penalty will be.
• To calculate the penalty, sentence is divided into chunks of unigrams, where each chunks contains the unigrams which are consecutively adjacent. So the longer the adjacent mapping the fewer the chunks. The penalty is given by
\[ p = 0.5 \left( \frac{c}{u_m} \right)^3 \]  \hspace{1cm} (13)
where \( c \) is the number of chunks and \( u_m \) is the number of unigrams that have found a mapping.

• The final score if given by
\[ M = F_{\text{mean}}(1 - p) \]  \hspace{1cm} (14)

• More robust metric is METEOR Universal which has a different more complicated way of calculating the mapping and thus the precision and recall. It was more linguistically involved and make use of function word list and paraphrase tables and has more free variables to tune to make the metric more transferable to other languages, which is not a requirement in our case as we resort to english for generating the sentences.

4.5 CIDEr: Consensus-based Image Description Evaluation

The link to the paper is [here](#).

4.5.1 Intro

• [Vedantam et al., 2015](#) proposed a novel approach to the evaluation of the quality of the descriptions generated for the image based on the human consensus.

• Their contribution was three fold. One they provide a new triplet based method to collect human annotations for capturing human consensus. Two, a automated metric which capture the consensus. Three, two new datasets PASCAL-50S and ABSTRACT-50S.

• The desirable properties one wants in the evaluation metric in the case of caption generation is grammaticality, saliency, correctness. We can measure these properties, but combining them into one metric is difficult.

• The metric discussed in the above section namely BLUE which is the precision-based metric is said to weakly correspond to the human judgment and so is the ROUGE metric which is recall based.

• CIDEr captures the consensus based similarity of the novel sentence with the already available human generated annotations. So the grammaticality, saliency and correctness are inherently captured.

• Most of the datasets apart from pascal50s and abstract50s, provide 5 human annotated sentences for each image. It is shown by the authors that 5 sentences are very small to measure how the majority of humans would describe an image. This lead to the MSCOCO dataset increase its sentences from 5 to 40 for its test portion.

4.5.2 Consensus Interface

• This section describes the human study protocol for generating ground truth consensus scores, which will be used to evaluate the proposed CIDEr metric.

• The annotators are shown three sentence say A, B and C and are asked which of the two B and C is most similar to A. No concept of similarity is given. B and C are the two candidate sentence and A is the reference sentence. This triplet is formed for all reference sentences of a image and for each choice of B and C. The relative nature of the task makes the assessment more objective.
4.5.3 CIDEr Metric

- Consider an image $I_i$, candidate sentence $c_i$, and image description set $S_i = s_{i1}...s_{im}$

- All words in both the candidate and reference sentence are first lemmatized and stemmed. Each sentence is represented with the set of n-grams.

- Now TF-idf weighting for each n-gram is done. The number of times an n-gram $s_k$ occurs in the reference sentence $s_{ij}$ is denoted by $h_k(s_{ij})$ or $h_k(c_i)$ for candidate sentence $c_i$. The TF-idf weighting $g_k(s_{ij})$ for each n-gram $w_k$ is given by

$$g_k(s_{ij}) = \frac{h_k(s_{ij}) \log \left( \frac{|I|}{\sum_{l \in I} \min(1, \sum_q h_k(s_{pq}))} \right)}{\sum_{w_j \in \Omega} h_l(s_{ij})}$$

where $\Omega$ is the vocabulary of all the n-grams and $I$ is the set of all images in the dataset. The first term measures the TF and second measures the idf.

- The $CIDEr_n$ score for the n-grams of length $n$ is computed using the average cosine similarity between the candidate sentence and reference sentence as it accounts for both precision and recall.

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\| \|g^n(s_{ij})\|}$$

where $g^n(c_i)$ is a vector formed by $g_k(c_i)$ corresponding to all the n-grams of length $n$ and same for $g^n(s_{ij})$

- The final CIDEr score for candidate sentence $c_i$ and reference sentence $S_i$ is given by

$$CIDEr(c_i, S_i) = \sum_{n=1}^N w_n CIDEr_n(c_i, S_i)$$

References


