

Proposing Contextually Relevant Quotes for Images

Shivali Goel*, Rishi Madhok*, and Shweta Garg*

Delhi Technological University, India

{shivali.bt2k14, rishi.bt2k14, shweta.bt2k14}@dtu.ac.in

Abstract. Due to the rise in deep learning techniques used for the task of automatic image captioning, it is now possible to generate natural language descriptions of images and their regions. However, these captions are often too plain and simple. Most users on social media and other micro blogging websites use flowery language and quote like captions to describe the pictures they post online. We propose an algorithm that uses a combination of deep learning and natural language processing techniques to provide contextually relevant quotes for any given input image. We also present a new dataset, QUOTES500K, with the goal of advancing research requiring large dataset of quotes. Our dataset contains five hundred thousand (500K) quotes along with the author name and their category tags.

Keywords: Computer Vision, Natural Language Processing, Automatic Image Captioning, Deep Learning

1 Introduction

Nowadays, as more and more pictures are uploaded to social media websites like Facebook or Instagram, the users often caption their pictures with deep meaningful quotes. Even after spending a lot of time and effort online, an appropriate quote may not be found. Therefore, we present a model which provides contextually relevant quotes for a given input image, thereby saving user’s time and suggesting better meaningful quotes that the user may skip while searching.

Previous work related to generation of simple natural language description of images includes that of [6] which was one of the first approaches to use Neural Networks for this task. Other approaches [1], [7] involved the use of an encoder decoder based architecture, similar to one used in machine translation to generate captions. The Neural network based models were further enhanced by adding an attention module [3], [12]. Prior to the use of neural networks, template-based description generation techniques were used [4], [5], [10]. Most existing work computing the similarity of two strings considers only syntactic similarities measures like the number of common words or n-grams [9], Jaccard Similarity [2], and Hidden Markov Model-based measure [11] etc.

To the best of our knowledge, this is the first work that delves into the combination of a quote search algorithm together with a neural image captioning system in order to suggest fancy captions for pictures.

* These authors contributed equally to this work.

2 Research Methodology

The proposed algorithm is divided into 3 parts. The first part takes as input, an image given by the user, and outputs a description of the image in natural language. The second part of the algorithm takes the description generated in the first part and then forms a list of candidate quotes, which are relevant for the image, using our QUOTES500K dataset. The third step measures the similarity between each candidate quote and the description generated in the first step. Finally, the three most relevant quotes are returned as the output for the input image. The architecture of the proposed algorithm is shown in Fig 1. Each of the steps are explained in detail in the following subsections.

2.1 Generating Natural Language description of Images

In the first step of the algorithm, we built an end to end neural network consisting of object detection and caption generation modules to generate captions. Output from the penultimate layer of the pre-trained model of VGG16 from Keras was used to extract features of the images in the dataset. This was fed into our image model consisting of two hidden layers. Output from this module was further passed on sequentially to the captioning module consisting of a LSTM layer which used words and image I as inputs and predicted one word of the caption at a time, considering the context of image observed and of the preceding words $p(S_t|I, S_0, \dots, S_{t-1})$. We trained the model using Stochastic Gradient Descent on the Flickr8K Dataset [13] for a total of 50 epochs reaching a loss value of 1.693. Batch size of 256 and categorical cross entropy loss function was chosen for our experiment. The caption so generated was given as input to the second step of the algorithm.

2.2 Proposing a set of Candidate Quotes

The second step of the algorithm filtered out and generated a list of candidate quotes i.e. quotes relevant to the image from our QUOTES500K dataset. To prepare this list, comparison was made recursively between two strings — the caption (fixed in all iterations) and a quote from the dataset until all quotes in it were compared.

Formally, let $C = \{c_1, c_2, \dots, c_n\}$ and $Q = \{q_1, q_2, \dots, q_m\}$ denote the caption and the quote string respectively. The initial steps included tokenizing the string, removal of stop words and stemming of the words, after which we were left with a bag of words, $bow = \{w_1, w_2, \dots, w_n\}$, for each string. We then applied the Part-of-Speech (POS) Tagger to extract the nouns from these bag of words. A word w_i was selected if $\forall i$, $POS(w_i) == NN | NNP | NNS | NNPS$.

Let $N = \{n_1, n_2, \dots, n_m\}$ denote the *noun_list*, which is the set of nouns for each string that were extracted from the previous step. Hence, N_C denotes the *noun_list* for the caption and N_Q denotes the *noun_list* for the quote. For each noun n_i in *noun_list*, we extracted the synonyms using the PyDictionary package and made a list of lists called the *syn_list* denoted by W_C for the caption and list of synonyms W_Q for the quote. Denoted by $W_C = \{\{c_{11}, c_{12}, \dots, c_{1p}\}, \{c_{21}, c_{22}, \dots, c_{2q}\}, \dots, \{c_{n1}, c_{n2}, \dots, c_{nr}\}\}$ and $W_Q = \{q_{11}, q_{12}, \dots, q_{1p}, q_{21}, q_{22}, \dots, q_{2q}, \dots, q_{n1}, q_{n2}, \dots, q_{nr}\}$.

The score s , was then calculated by searching if any word c_{ij} in W_C was present in W_Q , where c_{ij} is the j^{th} word for the i^{th} synonym list in the *syn_list*. If c_{ij} was present, the score was updated by one. An important point to note here is that if any one of the synonym word c_{ij} of the i^{th} list in W_C found a match in W_Q , the iterator was moved forward to the next list, $(i + 1)^{th}$ list of W_C , i.e. we did not look for a match for the other synonyms in the i^{th} synonym list. Mathematically, for each quote Q_k , the score s_k (initially $s_k = 0$) was given by,

$$\begin{aligned} \forall i, j \quad \text{If } c_{ij} \text{ in } W_Q : \\ s_k = s_k + 1 \\ i = i + 1 \end{aligned} \quad (1)$$

Then the quotes with the maximum score i.e. the quotes with score equal to the length of N_C , the *noun_list* for the caption, were listed out as relevant quotes in a list, denoted by *quote_list*, which was further passed as input to the third step of the algorithm.

$$\begin{aligned} \text{If } s_k == \text{len}(N_C) : \\ \text{quote_list.append}(s_k) \end{aligned} \quad (2)$$

2.3 Semantic Comparative Analysis of Vectorized Candidate Quotes

In the final step of the algorithm, each quote in the candidate *quote_list*, obtained from the previous step was encoded to a 4800 dimensional vector space. For this, we followed the approach of unsupervised learning of the sentence encoder described in [8]. The generic and distributed nature of the GRU (Gated Recurrent Unit) based encoder provided a perfect match for our application as we are not limiting our focus to a particular genre or domain while suggesting quotes.

Let candidate *quote_list* = $\{Q_1, Q_2, \dots, Q_n\}$, in which Q_i denotes a quote in this list and n is the total number of candidate quotes. Then the Vectorization Process is given by, $Q_i \rightarrow V(Q_i)$, where $V(Q_i)$ denotes the Vectorized form of Q_i and $V(Q_i) = [f_1, f_2, f_3, \dots, f_{4800}]$, where f_i is a floating point number. The Image Caption obtained from first step was also encoded to a vector using the same vectorization process, $C \rightarrow V(C)$ where C is the caption obtained from step 1.

Semantic relatedness between C and Q_i was then found by using Cosine Similarity as the similarity measure, $S(x, y)$

$$S(x, y) = \cos(V(Q_i), V(C_i)) = \frac{V(Q_i) \cdot V(C_i)}{\|V(Q_i)\| \cdot \|V(C_i)\|} \quad (3)$$

Other similarity measures were also implemented such as Jaccard Similarity, Euclidean Distance, Manhattan Distance, Minkowski Distance and Tanimoto Similarity. However, the best results were obtained by using the Cosine Similarity Measure. This step of our algorithm also explains the relevance of choosing to build a natural language caption generator in step 1 instead of a simple object detection module. This step further prunes the list of candidate quotes which contain key objects from the images(refer step 2) according to the semantic relatedness with caption generated in the first step.

Finally, the top 3 quotes from the *quote_list* having the maximum similarity measure were reported.

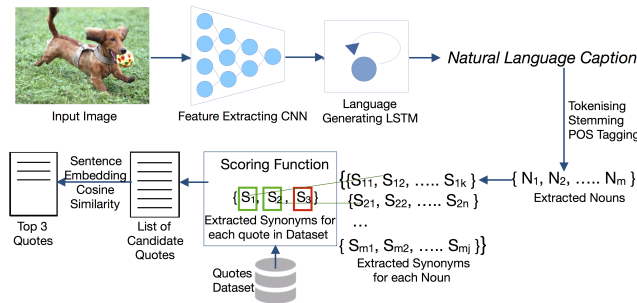


Table 1: Meta Data for Dataset

Attribute	Value
Quotes	499,709
Language Used	English
Total #Tags	7117
Avg #Tags/Quote	5
Min #Tags/Quote	1
Max #Tags/Quote	303
#Unique Authors	95,815

Fig. 1: Architecture of the Proposed Algorithm

Table 2: An example of a row entry in the Dataset

Quote	Author	Tags
A friend is someone who knows all about you and still loves you.	Elbert Hubbard	friend, friendship, knowledge, love

3 QUOTES500K Dataset



Since, there was no publicly available large dataset on Quotes, we prepared a dataset of our own. For this, we used the Python package — `BeautifulSoup`, to crawl quotes from various popular websites — Goodreads, Brainyquotes, Famousquotesandauthors and Curatedquotes. All quotes, except the ones in English, were removed using the Python package — `langdetect`. The final dataset prepared was in the `csv` file format which contains three columns — the quote, the author of the quote and the category tags for that quote. Examples of tags include — love, life, philosophy, motivation, family etc. These tags help in describing the various categories that a particular quote belongs to. The total number of quotes in our final dataset after crawling and further cleaning of the dataset was approximately equal to five hundred thousand (**500K**) quotes. We have made this dataset publicly available, so that it can be used by fellow researchers for educational and research purposes. The link to the dataset is given in the footnote ¹. The meta data for this dataset is shown in Table 1 and an example of a row entry in the dataset is shown in Table 2.

4 Results and Discussion

A total of 500 images from the Flickr8K Dataset were tested and provided with the top 3 quotes along with their similarity measures. Performance of the proposed algorithm was analyzed on the QUOTES500K dataset, using both qualitative as well as quantitative measures.

¹ Link to QUOTES500K Dataset

Table 3: Examples of results obtained (where +ve: Positive Evaluation and -ve: Negative Evaluation)

Image with Caption Generated	Top 3 Best Quotes	Similarity (in %)	Evaluation	
			+ve	-ve
 <p>a man is standing on a hill</p>	Change is the watchword of progression. When we tire of well-worn ways, we seek for new. This restless craving in the souls of men spurs them to climb, and to seek the mountain view.	77.3		
	Men trip not on mountains, they stumble on stones.	74.1	87	13
	Most of life is routine-dull and grubby, but routine is the mountain that keeps a man going. If you wait for inspiration you'll be standing on the corner after the parade is a mile down the street.	73.9		
 <p>a bird is flying through the trees</p>	But Hopes are Shy Birds flying at a great distance seldom reached by the best of Guns.	75.2		
	Perfect as the wing of a bird may be, it will never enable the bird to fly if unsupported by the air. Facts are the air of science. Without them a man of science can never rise.	73.9		
	It may be hard for an egg to turn into a bird: it would be a jolly sight harder for it to learn to fly while remaining an egg. We are like eggs at present. And you cannot go on indefinitely being just an ordinary, decent egg. We must be hatched or go bad.	72.8	81	19

4.1 Qualitative Analysis

The results of our proposed algorithm on two of the images, out of a total of 500 test images, are shown in Table 3. As seen from the table, the top 3 quotes provided for each of the two images display contextual coherency as well as semantic relatedness and hence can be classified as context aware quotes for these images. For instance, the top quote provided for the first image, having similarity measure 77.3%, beautifully expresses the meaning of the image. Even though, the second quote for the same image is not as good as the first one, but still the nouns, "mountain" and "man" are retained by our model and the quote displays contextual coherency.

4.2 Quantitative Analysis (User Study)

To test the usefulness of our results, a user study of 100 users was conducted. To avoid the bias in our evaluation, the users were deliberately selected across various age groups ranging from 10 years to 60 years. Each user was presented with a web portal wherein the

user was shown an image and the top 3 most relevant quotes for that image. The user was then presented with a choice to rate the quotes as either "Relevant Quotes" or "Irrelevant Quotes". This experiment was performed for 500 images and for each image, 100 users evaluated the quotes generated by the model. Table 3 shows the results obtained from our model for 2 images. The overall accuracy obtained for our model was 83.2 % which was calculated as the average positive response for the 500 test images.

5 Conclusion

In this paper, we presented a dataset of approximately 500K quotes which would significantly advance the research in providing Context Aware Quotes for Images. Furthermore, an algorithm was proposed which provides contextually relevant quotes for an image. The results obtained from our algorithm were evaluated by using both qualitative as well as quantitative measures.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014)
2. Chaudhuri, S., Ganti, V., Kaushik, R.: A primitive operator for similarity joins in data cleaning. In: 22nd International Conference on Data Engineering (ICDE'06). pp. 5–5 (April 2006)
3. Denil, M., Bazzani, L., Larochelle, H., de Freitas, N.: Learning where to attend with deep architectures for image tracking. *Neural Comput.* 24(8), 2151–2184 (Aug 2012)
4. Elliott, D., Keller, F.: Image description using visual dependency representations. In: EMNLP. pp. 1292–1302. ACL (2013)
5. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every Picture Tells a Story: Generating Sentences from Images, pp. 15–29. Springer Berlin Heidelberg (2010)
6. Kiros, R., Salakhutdinov, R., Zemel, R.: Multimodal neural language models. In: Xing, E.P., Jbara, T. (eds.) *Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 32, pp. 595–603. PMLR, Beijing, China (22–24 Jun 2014)
7. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. CoRR abs/1411.2539 (2014)
8. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R., Fidler, S.: Skip-thought vectors. CoRR abs/1506.06726 (2015)
9. Kondrak, G.: N-Gram Similarity and Distance, pp. 115–126. Springer Berlin Heidelberg (2005)
10. Li, S., Kulkarni, G., Berg, T., Berg, A., Choi, Y.: Composing simple image descriptions using web-scale N-grams, pp. 220–228 (2011)
11. Miller, D.R.H., Leek, T., Schwartz, R.M.: A hidden markov model information retrieval system. In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 214–221. SIGIR '99, ACM (1999)
12. Mnih, V., Heess, N., Graves, A., kavukcuoglu, k.: Recurrent models of visual attention. In: *Advances in Neural Information Processing Systems 27*, pp. 2204–2212 (2014)
13. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using amazon's mechanical turk. In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. pp. 139–147. CSLDAMT '10, Association for Computational Linguistics (2010)