Facilitated Deep Learning Models for Image Captioning

Imtinan Azhar
Computer Science Department
University of Sharjah, UAE
iazhar@sharjah.ac.ae

Imad Afyouni
Computer Science Department
University of Sharjah, UAE
iafyouni@sharjah.ac.ae

Ashraf Elnagar
Computer Science Department
University of Sharjah, UAE
ashraf@sharjah.ac.ae

2021 55th Annual Conference on Information Sciences and Systems (CISS) | 978-1-6654-1268-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/CISS50987.2021.9400209

***Abstract*—This paper focuses on developing semantic image caption generation techniques that leverage image and scene understanding. More particularly, we are interested in addressing image captioning by developing a mixture of object detection and attention-enriched deep learning models. To extract the image features, a Convolutional Neural Network (CNN) is used, and then an extended version of Recurrent Neural Networks (LSTM) with attention-enrichment is adopted to generate the caption. We implement image captioning by considering detected objects from the image scene, and then by integrating an attention mechanism for caption generation. This can have multiple advantages from accuracy and semantics perspectives. The objective of this paper is to introduce a combined pipeline that employs several variant models for semantic caption generation. Four variant models are proposed, all of them are implemented and trained on COCO and Flickr30k datasets, and then tested on a subset of COCO dataset. Results of the different models were evaluated using a semantic similarity analysis between the generated captions and the actual ground truth captions. Our framework helps in a deeper understanding of images and decision making in diverse use-cases such as innovative and distinctive responses from multimodal data, and in analyzing and monitoring crowdsourced images from social media and other sources.**

***Index Terms*—Image captioning, deep learning, attention-enrichment, object detection, semantic similarity**

I. Introduction

Recently, image captioning is becoming an active subject of research attracting great interest from researchers. This is due to the fact the field combines domain expertise from two largely researched fields in Artificial Intelligence, namely, Natural Language Processing and Computer Vision. The task in essence is to caption images using artificial intelligence, where a human understandable sentence is produced by a machine when faced by an image, said sentence is an accurate description of the content of the image. This task can help in a number of practical applications, including search engine optimization, or image retrieval tasks, among others. Despite the large number of practical use cases, the task gathers its fame from its complexity. The multimodal nature of the task puts strain on the training process of traditional neural networks. Neural networks have to not only learn the process of image feature extraction but also to encode language heuristics into the generation process that produces the caption. Learning these multimodal features can be a taxing task for a network.

Although many studies have been presented recently to address these challenges, there is still lack in producing highly semantic captions that closely and accurately describe the image content [1]. We propose a new hybrid model that facilitates the image captioning process, by generating a description based on detected objects and important regions from the image scene. Our approach takes away some computer vision responsibilities from the image captioning model to help facilitating the generation process, resulting in much more accurate generations. Our contributions to the paper are as follows. We introduce multiple variants of facilitated captioning models that will semantically evaluated against the ground truth captions. These models include: i) the object-based captioning model with possibility to detect multiple objects per scene; ii) an attention-focused model; iii) a facilitated attention-enriched model that prioritizes objects in the formulated sentences based on the focused zones detected by the attention mechanism. We also produce a semantic evaluation pipeline that makes use of sentence encoders to evaluate generated captions.

In the remainder of the paper, we will detail different sections. Section II discusses the related work in image captioning. Section IV our methodology for model development and training. Section V introduces our process for end-to-end caption generation; while Section VI-C details the evaluation process. Finally, a concluding section that highlights our future work and overall insights.

II. Related Work

For early approaches like template-based approach, Kulkarni et al [2] proposed to extract the image attribute tuples (object, visual attribute, spatial relationships), and then generate words by using n-gram based language models to retrieve the final caption. However, such approaches can result in having noisy estimations of the visual content and poor alignment between images, and additional re-ranking procedures may be needed based on textual features in order to improve caption accuracy.

A. Deep Learning Approaches

Deep learning methods became widely used in image captioning, since it can generate novel captions by analyzing the visual content of the image using an image model, then generate the caption using a language model. Deep learning made a significant progress in the fields of CV and NLP

978-1-6654-1268-1/21/$31.00 ©2021 IEEE

techniques, making it achieve state-of- the art results for image captioning tasks [1]. For instance, the survey conducted by Zhong et al. [3] shows the importance of object detection in understanding image content and in extracting descriptive semantics of an image. A study carried out in [4], explores the relationship between objects and image captioning. This study promotes the usage of objects in cohesion with captioning models and strengthens our confidence in our proposed model.

Related work on deep learning for image captioning usually uses an encoder-decoder architecture with multiple variants that introduce improvements at the encoder, decoder or at both levels [1]. An Encoder-Decoder architecture is inspired by the machine translation model proposed by Sutskever et al. [5]. It mainly reads the source sentence and converts it into a rich fixed-length vector representation, which is used as the initial hidden state of the decoder to generate the target sentence. The extracted feature vector are fed to the decoder LSTM to generate the caption word by word. Although neural encoder-decoder approach proposed in [6] is one of the effective methods in image captioning, it cannot analyze the image over time while generating the caption, and does not focus on specific parts of the image. Instead, it generates the caption by considering the scene as a whole. Xu et al. [7] was the first to propose an extended version of the basic encoder-decoder approach, which is an attention-based model that focuses on the salient parts of the image dynamically and generates the corresponding words during decoding process. Other studies on the effects of attention in image captioning have promoted its usage quite heavily [8]–[10], by measuring the ground truth for attention map and the labeling of image regions for both Flickr30k and MS COCO datasets.

B. Language Models: RNN / LSTM / Others

Many approaches were used to perform sequence to sequence learning tasks such as log-bilinear models, skip-gram models, and recurrent neural networks (RNNs). RNN is used in many sequence learning tasks such as machine translation, speech recognition and image captioning tasks [11]. Unlike feed forward networks which go in one direction and do not form a cycle, RNN's nodes can form a directed graph, and use their internal memory to process sequences of inputs, by passing their current input with the previous hidden state, making it more applicable to sequence learning tasks. Traditional RNN suffer from vanishing and exploding gradient problem, which means that they cannot predict words in long-range dependencies. Therefore, LSTM, an improved version of RNN is adopted in many studies. An LSTM has special units in addition to the standard units of RNN that uses a memory cell that maintains information in memory for long periods of time and decides what to keep and what to forget [12].

Vinyals et al. [6] used the LSTM as a decoder for the encoded image to generate the caption. Despite the wide use of LSTM to generate captions, this type of RNN requires a significant amount of storage due to its long-term dependency to the memory cell that keeps information for a long period of time. Therefore, CNN was proposed for sequence to sequence learning tasks, since it is faster in processing than LSTM, and can learn the ignore underlying hierarchical structure of a sentence [11]. A recent approach proposed an enhanced LSTM architecture that considers the semantic roles of words towards a better sentence modeling [13]. Their results shows a better performance in terms of learning speed and semantic scores.

Our approach benefits from the cutting-edge techniques and approaches, and proposes an intelligent hybrid deep learning model that drives the caption generation by considering the significant objects in the image scene in order of their important, and by employing attention-enrichment for achieving a highly semantic description of the image.

III. Methodology

We propose a facilitated captioning model that leverages the power of object detectors in cohesion with predefined heuristics, feature extractors and attention-enriched language models to generate semantic captions. This end-to-end system can be visualized in Figure 1. The entire system is a composition of multiple subsystems, each responsible for its own set of tasks (i.e., detection and feature extraction). In our proposed model, the system takes in an image, which is first processed and then passed on to the the object detector in an appropriate format. The detector returns a list of detected objects, which then undergo some predefined heuristics to generate a starting point for our captioner. The captioner is a mixture of a feature extractor and a language model with attention enrichment. This collective sub-system is responsible for taking in an image, extracting relevant features from it using the feature extractor, and then passing the extracted features along with an initial starting point to the language model, where it iteratively generates a caption. The initial starting point, in this case, is the one that our heuristics generate using detected objects. This starting point is appended to at each iteration of the language model. This addition may also contain objects from the object detector. The positioning and the presence of said objects in the current state are identified by the aforementioned predefined language heuristics. The final combined state at the end of the iterative cycle, which is identified by a special token, is the generated caption.

In our proposed model, we facilitate the captioner with pre-generated starting points, the reason being that we facilitate certain information towards the language model. For example, if we detected a person in an image, now instead of having the starting marker as a starting point, we could generate a starting point as *'¡starting marker¿ a ¡detected object¿'*. The captioner would then be responsible for generating the caption here on after. This in turn makes us less reliant on the captioner itself for the understanding of objects in the image and allows us to distribute responsibilities / accountability across networks. We would now rely on the object detector for extracting objects from the image as opposed to overburdening the captioning model and in return creating a single point of failure.

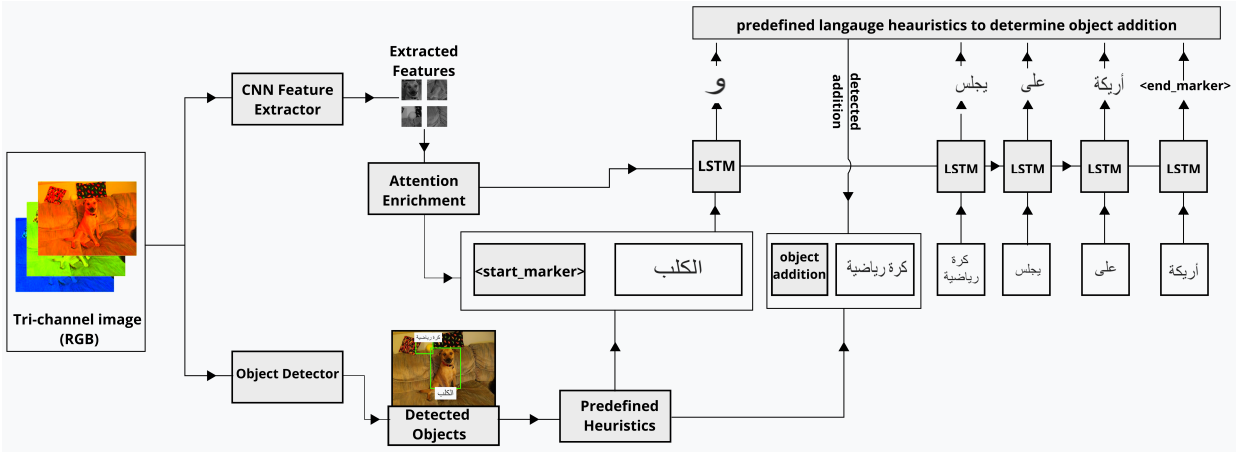

Fig. 1. Object-based, attention-enriched captioning architecture

IV. Model Architecture and Training

The training happens in a multi-staged manner. Different modules need to be trained for distributed inference and training purposes, in order to compile the final solution. The training can be divided into 3 stages and the specific architectures will be described accordingly.

A. Pretrained Feature Extractor

The first stage of the model is to use a pretrained feature extractor that is able to extract image features from a given image. Our network backbone was chosen to be an ImageNet pretrained ResNet50 model. This selection was made after comparisons with pretrained Inception v3 and VGG16 models. ResNet50 was selected from this bucket for a multitude of reasons that, if not considered, would not negatively affect the performance, but would rather hinder the development process. Although it should be noted that any one of these models would provide a satisfactory performance. For the purpose of this experiment, a pretrained model was more than satisfactory to provide preliminary results, no transfer learning was performed and the models were used as is, as provided by libraries such as Tensorflow. The classification head of the model is stripped so the network now outputs a feature representation of an image as opposed to a classification label. This takes care of stage one of the architecture.

B. Object Detection: Architecture and training

The second stage of the model requires the development of a detection module. We followed suit with the theme of simplicity and went with predefined architectures that are proven to work. A RetinaNet model with ResNet50 as its backbone was used as our object detector. The model was trained on the COCO dataset for object detection. The detector was responsible for detecting objects belonging to one of seventy seven classes. It should be noted that other denser datasets (those with higher number of classes like Cifar100 or the Imagenet datasets) could have also been used, but for the demonstration of the proposed model and in lieu with our theme of simplicity this dataset was more than satisfactory. The model was trained until a reasonably mean Average Precision (mAP) of 0.45 was achieved and the model performance was deemed good enough for the demonstration of our proposed model.

C. Captioning Language Model: Architecture and training of the language model

The third and final stage of the model required the development of a language model that would be responsible for taking in the learned image features and to generate a caption from said features. This model would be required to identify objects in an image, along with language semantics and structures and to output a comprehensive human understandable description of the image content. Various model architectures were implemented and evaluated. Experiments were run on a variety of different models, from simplistic tri-layered LSTM models to much more complex attention enriched language ones, a number of experiments were conducted to chose the optimal model for this purpose. Due to the variation of architectural nature and performance capabilities of such models, two models were chosen to demonstrate performance capabilities of our proposed solution. The first was the tri-layered LSTM model (which we will call the 'simplistic language model' through the remainder of this paper) and the second was the attention enriched language model. It should be noted that our experiments also made use of variations of the COCO dataset and the original version with captions, and variations of the Flickr30k datasets for training purposes.

Image features needed to be extracted to train the two language models, hence for performance purposes, a one-time extraction of all image features was done, and these representations were saved to disk to avoid re-evaluation of the same samples across different epochs and different model training procedures. Each model was then trained over a pre-processed version of the captions. The two models were trained while varying hyper-parameters (i.e., the learning rate / number of nodes per layer / number of layers / number of

epochs) until satisfactory enough performance was reached. It should be noted that although the simplistic model was quicker to train in terms of seconds per step, the overall convergence of the model was slower than that of the attention enriched model.

Algorithm 1 Caption generation process

```
caption = [ ]
img = preprocess(img)
features = featureExtractor(img).output
objects = objectDetector(img).output
prediction = indexOf(startMarker)
while prediction ≠ indexOf(endMarker) do
    caption.add(wordOf(prediction))
    caption = addObjects(caption, objects) [1]
    prediction = predictNextWord(features, caption)
end while
caption = caption.joinWith(' ')
caption.remove(wordOf(startMarker), ifExists)
return caption
```

V. Staggering Stages: end-to-end captioning

The stages are then staggered to create an end to end pipeline for caption generation. The object detection and feature extraction stages are stacked to run in parallel, both stages receive the image and produce different outputs. This can be done easily since the two stages do not have any inter-dependencies. The language modeling stage is staggered so that it runs after both previous stages have finished completion; this is necessary since the language modeling stage requires the output of both prior stages to run. This end-to-end system is responsible for generating captions and can be observed in Algorithm 1. Addition of objects is subject to multiple predefined heuristics, hence the returned object is a modified version of the input caption or it may be the exact original caption itself.

A. Caption Generation

This section details the workings of the pseudo-code presented in Algorithm 1. The algorithm expects an image as its input of size $(w, h, 3)$ and outputs a caption that represents contents of the input image. *Lines 3-4:* The input image is forwarded to the object detector and feature extractor simultaneously. The object detector outputs a list of observed objects, and the feature extractor outputs a learned representation of an image. This feature set, once acquired, can be used to generate captions. *Line 5:* Language specific heuristics are then applied as a post processing step for the language modeling stage, these predefined heuristics make use of the extracted list of objects. Objects of focus are identified and count frequencies are extracted. These heuristics help to generate a starting point to pass onto the language modeller. *Lines 6-10:* The generated starting point along with the extracted features are then passed onto the language modeller in multiple sequential passes. At each pass, the output of the model is appended to its input state. Additional language specific heuristics are applied onto this state; it is evaluated if the object detection can make contributions to the current state, if yes then the state is further modified. This modified (or unmodified) state, is then passed on to the next sequential model pass. This cycle of state update and modification is continued until an ending point is observed. At this point, the final input state is taken as is. Any network specific additions that were made (starting point tokens etc.) are then stripped off, and the cleaned state is now treated as the generated caption. *Lines 11-12:* Language specific heuristics are large in number and vary in terms of complexity, some are as simple as certain pronoun detection, at which point it is observed that object additions need to be made and objects of focus are manually added. Other heuristics are a bit more complex and require count frequencies to modify current input to accommodate multiple instances of some objects (if they are treated as singular). It should be noted that the predefined heuristics make some assumptions, such as the object of focus, and/or number of objects to use (not all objects are used). To circumvent this, several captions can be generated, where each caption would be created with different object sets. These captions can then be evaluated with the original set to determine the accuracy of each generation set.

B. Caption Generation with Attention Enrichment

Attention plays an important role in training the neural networks. It helps the neural networks converge faster and has also proven to give superior performance when compared to similar models without the mechanism. Attention mechanisms traditionally help by allowing neural networks to focus on certain regions of an image, as opposed to the entire image itself. This allows the network to focus on what is important and not to be disrupted by background noise that may or may not be present. The term $background\ noise$ has been loosely used to describe certain regions of the image that may take focus away from regions of an image that are more important. What matters in attention mechanisms is how the regions of interest are chosen. This is done by making use of context. In the initial stages of caption generation, attention mechanisms play a small role since there is a little to no context to chose the appropriate regions. Attention kicks on in the latter stages when context starts building up. This introduces room for error, since if the context is irrelevant to begin with then the selected regions of interest may not be desirable to begin with, hence the generations could be flawed from the get-go.

This is where our proposed facilitated methodology shines. By using our self-generated starting point we not only facilitate the language model, but also facilitate the attention mechanism that it makes use of. The initial irrelevance of the attention mechanism is removed, by now utilizing the facilitated input as its context, the attention mechanism can now provide much more accurate regions of interest from the very start of the generation process. We hypothesize that this eventually helps us in achieving much more accurate captions, and this

hypothesis is verified by our results that can be viewed in the evaluation section.

VI. Evaluation of the Proposed Methodology

We hypothesize that our effective model can improve captioning performance using object detection and attention enrichment for facilitated generation. To actually determine if our facilitated networks give better performance in comparison to their original counterparts, an evaluation pipeline is set up. Our research experiments with unique set of models, one using a simplistic language model, and the others based on object detection and attention enrichment. These models had been altered to work with our facilitated methodology, though meanwhile their original counterparts still existed, bringing the total number of models to four. These models were evaluated on pre-captioned sets of images, by comparing the similarity of the generated captions of each model with the ground truth captions. The goal of this evaluation is to determine if the facilitated model is able to provide semantically closer captions when compared to their original counterparts.

A. Computing Similarity

To compute similarity we will be using cosine similarity. Cosine similarity is a metric used to measure the similarity between two documents. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space, the smaller the angle, the higher the cosine similarity. To compare different sets of captions we need the aforementioned projected vectors for each caption, for these purposes we use the Universal Sentence Encoder (USE). USE encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. The pretrained Universal Sentence Encoder was used, it was taken from the publicly available Tensorflow-hub repository. Our various generators are used to generate captions, then the vector representations for both generated captions and the ground truth captions are compared using the universal sentence encoder. We then use cosine similarity on all extracted vectors to compute the similarity sim between the generation vectors $V_{generated}$ with the ground truth vectors V_{actual}.

$$sim(V_{generated}, V_{actual}) = V_{generated} \cdot V_{actual}$$

B. Scoring the results

For each image in our evaluation set the similarity scores of each generator are stored and then the generators are ranked. The rank on a single evaluation image i for generator n is the summation of the scores of the generator when compared to the similarity scores of other generators. The generator is assigned a score of 1 when the similarity score sim_n^i of generator n on image i is greater to the similarity score sim_g^i of another generator g on the same image, given that $n \neq g$, otherwise the generator is assigned score 0. This scoring is done against all generators G. It should also be kept in mind that ranking is only awarded when the similarity sim_n^i has crossed a certain threshold t, otherwise the specific generator assigned an overall is rank 0. This method ranks all the generators, giving the highest rank to the best performing generator, and giving the lowest rank to the worst generator(s), given that all generators pass a certain mark of quality.

$$rank_n^i = \begin{cases} \sum_{g=0}^{G} \left[\begin{cases} 1 & \text{if } sim_n^i > sim_g^i \\ 0 & \text{else} \end{cases} \right] & \text{if } sim_n^i \geq t \\ 0, & \text{otherwise} \end{cases}$$

The overall score for generator n can then be calculated by summation over the calculated ranks $rank_n^i$ for a generator over all images I. The generator that performed the best across majority of the images had the highest overall score and the worst generator had the lowest.

$$score_n = \sum_{i=0}^{I} rank_n^i$$

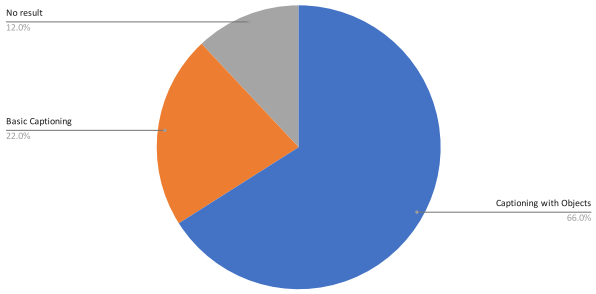

Fig. 2. Percentage of the times each generator outranked the other

C. Evaluation datasets and results

Our evaluation dataset were randomly sampled images from the Flickr30k data set. Roughly a total of 450 images were tested on, all images were passed through each of the 4 models, the score for all generators were calculated by summing the rank of each generator across all images. The goal of the evaluation was to see if the facilitated generators performed better than their unfacilitated counterparts. These results can be seen in Figures 2 and 3. The charts compare the facilitated and unfacilitated variations of each generator model, the simplistic captioning model and the attention based captioning model.

It can be seen from Figures 2 and 3 how facilitated generators gave superior performance to their unfacilitated counterparts. For the basic generator it could be seen that for around 12% of the images in the evaluation set, both generator versions failed to provide good quality captions (meaning both the generators were unranked due to failure to clear the quality threshold), it was observed that the facilitated captioner gave superior performance on around 66% of the images, which is around ⅔ of the evaluation set. Also, around 75% of the set for which at least one of the captioner was able to provide a result, meanwhile the unfacilitated was only able to beat the facilitated version in 22% of the tested images, thus it can be

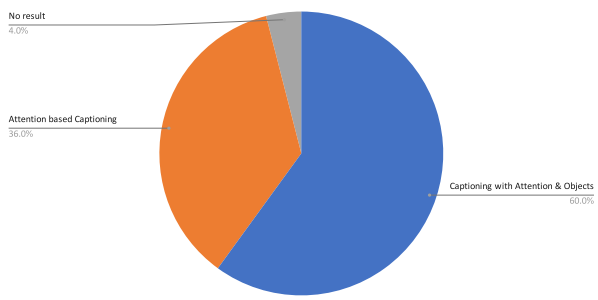

Fig. 3. Percentage of the times each generator outranked the other

seen that the facilitated network was able to provide superior performance for the basic models.

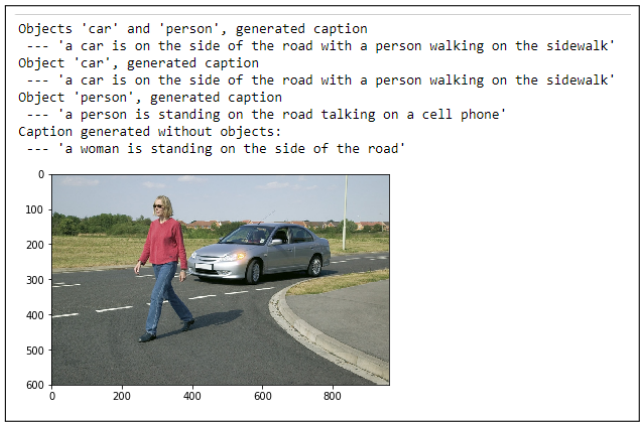

Fig. 4. Generations over multiple objects

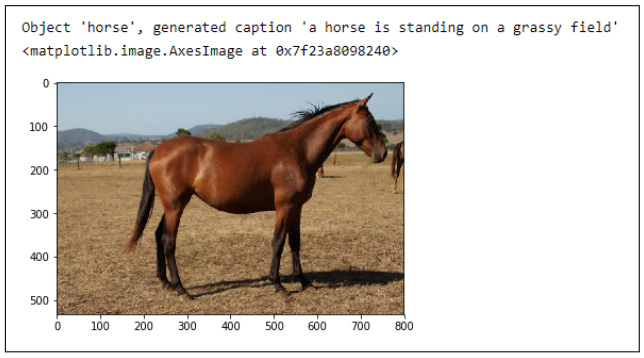

Fig. 5. A sample generation using the facilitated network

A sample of the generations can be seen in Figure 5 for a generic generation, and Figure 4 showing the different captions generated with focus on different objects. The attention based generators were a bit more competitive, so much so that for only 4% of the images both generator versions failed to provide good quality captions. It could once again be observed that the facilitated generator gave superior performance to its unfacilitated variation. For around 60% of the images (63% of the set for which at least one of the captioner was able to provide a result), the facilitated model beats the unfacilitated one, with the unfacilitated once going ahead in only 36% of the tested images. Though the unfacilitated attention based model gave a tougher competition to its counterpart in comparison to the unfacilitated basic model, it still could not manage pulling out an overall superior performance.

VII. Conclusion

We presented the design and implementation of three deep learning models for facilitated image captioning, by considering object recognition and/or an enhanced attention mechanism to automatically generate image captions. Results of the different models were evaluated using a semantic similarity analysis between the generated captions and the actual ground truth captions. Our evaluation experiments demonstrates that facilitated image captioning (regardless of different architectures) can help provide superior performance to their unfacilitated counterparts. Using the facilitated methodology allows us to distill responsibility as well, removing a single point of failure. In the future further experiments can be run with a various number of changes, which include, but is not limited to using much densely trained detectors, using larger captioning datasets, using different architectures for the generative language models.

References

[1] Y. Wang, J. Xu, Y. Sun, and B. He, "Image captioning based on deep learning methods: A survey," *arXiv preprint arXiv:1905.08110*, pp. 1–7, 2019.

[2] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: Understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[3] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[4] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *European Conference on Computer Vision*. Springer, 2018, pp. 711–727.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, 2014, pp. 3104–3112.

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[7] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[8] C. Liu, J. Mao, F. Sha, and A. Yuille, "Attention correctness in neural image captioning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 4176–4182.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[10] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 4633–4642.

[11] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[12] Q. Wang and A. B. Chan, "CNN+ CNN: Convolutional decoders for image captioning," in *31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, 2018, pp. 1–9.

[13] J. Kleenankandy and A. N. K A, "An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies," *Information Processing & Management*, vol. 57, p. 102362, 2020.