Proceedings of the Federated Conference on Computer Science and Information Systems pp. 129–136
DOI: 10.15439/2020F57
ISSN 2300-5963 ACSIS, Vol. 21

Exploration into Deep Learning Text Generation Architectures for Dense Image Captioning

Martina Toshevska*, Frosina Stojanovska*, Eftim Zdravevski*, Petre Lameski* and Sonja Gievska*
*Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University - Skopje, North Macedonia
Email: {martina.toshevska, frosina.stojanovska, eftim.zdravevski, petre.lameski, sonja.gievska}@finki.ukim.mk

***Abstract*—Image captioning is the process of generating a textual description that best fits the image scene. It is one of the most important tasks in computer vision and natural language processing and has the potential to improve many applications in robotics, assistive technologies, storytelling, medical imaging and more. This paper aims to analyse different encoder-decoder architectures for dense image caption generation while focusing on the text generation component.**

Already trained models for image feature generation are utilized with transfer learning. These features are used for describing the regions using three different models for text generation. We propose three deep learning architectures for generating one-sentence captions of Regions of Interest (RoIs). The proposed architectures reflect several ways of integrating features from images and text. The proposed models were evaluated and compared with several metrics for natural language generation. The experimental results demonstrate that injecting image features into a decoder RNN while generating a caption word by word is the best performing architecture among the architectures explored in this paper.

I. Introduction

Describing images, also known as image captioning, is the process of generating a textual description that best explains the image scene. Automatically describing the content of an image is a problem in artificial intelligence that connects computer vision and natural language processing. The textual description is expected to represent not only the presence of objects but also the interaction between them, as well as their characteristics and relationships [1], [2], [3].

Recognizing and describing the content of images is a very important task in many applications, including assistance to people with visual impairment (e.g., for text-to-voice guidance), robotic systems, vision-based search engines, and more. For most applications, the image captioning system must give an accurate description of the scene [4]. Additionally, image captioning can be used for automated scene description and its output can be used for automated training of models for other domains, such as other assistive technologies, storytelling, medical imaging, health-care, behaviour analysis, visual surveillance, and more.

Generating caption for a given image requires a strong understanding of its content. With the rise of deep learning techniques, understanding the content of an image relies upon convolutional neural networks. Detecting objects, as well as their properties and relations, is the primary concern for caption generation. Object detection is a widespread research area which comprises of many well-performing models [5].

The problem of automatically describing images can be split into two sub-problems: understanding the content of the image, which is considered as a computer vision task, and generating text sequences, a natural language processing task. Various approaches are used in the area of object detection, and the most successful ones are based on deep learning techniques. Models like R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], Mask R-CNN [9] utilize region proposal networks (RPNs) to detect objects in an image. On the other hand, the method described in [10], often referred to as VGG, named by the group that proposed it, focuses on classifying the image scene with Very Deep Convolutional Networks.

The problem of image captioning can be split into two main approaches: (1) generation of a single description of an image, and (2) describing different Regions of Interest (RoIs) from a single image, also known as dense captioning [4]. The dense image captioning describes several regions of the image that contain objects and some relations between them. Therefore, the problem is considered as a more informative strategy when describing images, but also a more difficult one.

Concerning the problem of image captioning, many researchers are using hybrid deep learning models, that is, a combination of a convolutional neural network (CNN) and a recurrent neural network (RNN). The models developed for object detection, RoI proposal, image segmentation, and related problems are achieving great performances [11]. These models are used as feature extractors of images and specific regions in the images. The main question is, can we use the already designed models as feature extractors, and then describe the regions in an image with models designed for text generation.

To answer this question we conducted experiments with three deep learning architectures for generating text captions of RoIs in the image[1]. We apply a transfer learning approach using a pre-trained object detection network from Mask R-CNN for determining RoIs and their corresponding features. The integration of features describing images and the context of previously generated text was performed using three different models for text generation. We evaluated and compared

[1]The code for this research is available at https://github.com/frosinastojanovska/image-captioning

the models using well-known evaluation metrics for natural language generation. The discussion of the performance of the models is introduced along with the evaluation results.

The rest of the paper is organized as follows. Section II reviews the relevant related work. Section III describes the utilized dataset, the proposed architecture for extraction and captioning of regions of interest (RoIs) in images and the used evaluation metrics. Next, in Section IV, we present and discuss the results of our experiments. Finally, Section V concludes the paper and identifies directions for future research.

II. Related Work

There is an abundance of models introduced in the domain of image captioning, originating from the models that generate single image caption to models that generate multiple captions for an image. The first group includes models that generate a single caption for the whole image [12], [13], [14], [15].

To describe the entire image with one sentence, the NIC approach [12] uses a CNN pre-trained for an image classification task. This method encodes images into a compact representation, followed by an LSTM network that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence for a given image, which is fed into the LSTM only once.

Attention-based models focus on a specific image part (i.e., region or object). A visual attention-based model with hard and soft attention alternatives is proposed in [13]. As the model generates each word, its attention changes to reflect the relevant parts of the image. A semantic attention-based model is proposed in [14]. This model learns to selectively attend to semantic concept proposals and fuse them into hidden states and outputs of RNNs. The selection and fusion form feedback combining a top-down approach, which starts from a gist of an image and converts it into words, and a bottom-up approach, which combines words describing various aspects of an image.

The model presented in [15] consists of object detection and localization model to extract the information of objects and their spatial relationship, and RNN with attention mechanism to generate sentences. The encoder first uses Faster R-CNN to detect objects and then applies VGG to create feature representation for detected object regions. Captions are generated with an LSTM conditioned on the attention of the detected object regions, previously generated tokens and a previous hidden state.

Recent approaches [16], [3], [17], [18] incorporate the Transformer [19] architecture instead of traditional RNNs for caption generation. The underlying architecture remains Encoder-Decoder, but the structure differs from previous CNN-RNN approaches. Faster R-CNN [8] is used as image encoder in [17], [3], ResNext [20] in [16], and a novel Image Transformer in [18]. For all methods, Transformer is applied as a decoder to generate the caption.

The second group consists of models intended for dense image captioning. These models, unlike the models described above, generate a caption for each region of an image. The DenseCap model [21] achieved exceptional results in describing image regions. It consists of convolutional and recurrent networks responsible for detecting RoIs and their vector representation, respectively. DenseCap is a convolutional localization layer based on VGG similar to the one applied in Faster R-CNN with several modifications. The localization layer identifies spatial regions of interest and extracts a fixed-sized representation from each region. The second part is an LSTM for creating descriptions.

Another approach for dense captioning is presented in [22]. It relies upon Faster R-CNN for region features extraction. This model is an improvement of the DenseCap model. The improvement is two-fold: (1) incorporate global context feature of the image, and (2) late fusion of the region features. Authors in [23] present a Multimodal RNN that uses visual-semantic alignments. This alignment method is based on a combination of a CNN that processes image regions, a bidirectional RNN that processes sentences, and a structured objective that aligns the two modalities through a multimodal embedding.

Novel approaches [24], [25] rely upon object context features for generating a caption. CAG-net [24] uses Faster R-CNN for region extraction and custom contextual feature extraction for extracting features of the target region as well as a global feature of the whole image and features of neighbouring regions. The features are then fused and fed into an LSTM network to generate region caption. Another approach presented in [25], proposes two different architectures. The first architecture, COCD, uses an LSTM to decode the object context. It is then concatenated with caption LSTM in order to generate the final description. In the second architecture, COCG, the object context is fed into caption LSTM as guidance information for generating the region description. For both architectures, the object context is obtained with gLSTM module [26] with region features, extracted with Faster R-CNN, as guidance information.

In this paper, we focus on the caption decoder of an encoder-decoder based architecture for dense captioning. We employ the Mask R-CNN module [9] for region extraction with a transfer learning approach. We explore different architectures for decoding region features into region captions.

III. Methods and Analysis

A. Dataset

In the experiments, we used the Visual Genome dataset [27], consisting of 108,077 images with 5,408,689 region descriptions. An exemplary image with three regions is shown in Fig. 1. Each image region (i.e., a RoI) is described with the following parameters: width, height, x coordinate, y coordinate, and caption. The distribution of the number of regions per image and caption length is shown in Fig. 2.

B. Feature extraction based on Mask R-CNN

The dense image captioning is the problem of generating descriptions of RoIs in an image. Therefore, the RoIs need to be extracted from the image and described with a fixed-length feature vector. This vector then is an input into another part of

Fig. 1: Sample image, regions of interests and their corresponding captions from Visual Genome [27]

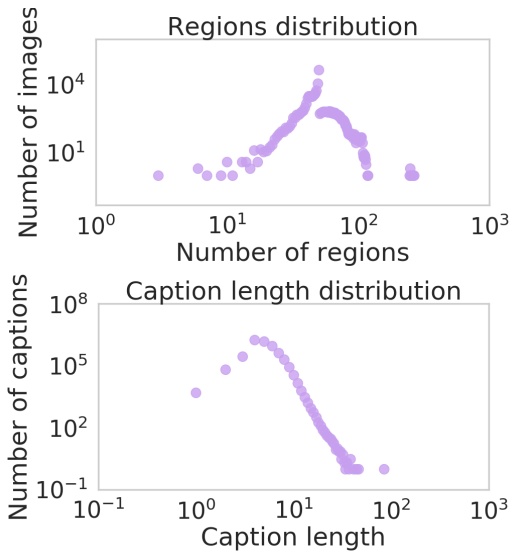


Fig. 2: Distribution of the number of regions per image (top) and caption length (bottom)

the model for text generation. There are several deep learning convolutional models for this problem. The R-CNN [6] model is improved with the next version of the Fast R-CNN [7] that facilitate feature extraction from RoIs with any dimension into a fixed-sized feature vector. Then, the Faster R-CNN [8] is the next improvement that adds a Region Proposal Network (RPN) for detection of RoIs which are fed to the Fast R-CNN model.

Mask R-CNN model [9] is a method for object detection and segmentation. It extends the Faster R-CNN [8] model by adding a new branch for mask detection and introducing RoIAlign technique. RoIAlign is a modification of the RoIPool technique [7], which extracts a feature map with quantization from each RoI. RoIAlign replaces the quantization with bilinear interpolation aligning the extracted features with the RoIs.

In this paper, to generate the RoI feature representations, we utilize the first stage of the Mask R-CNN model (RPN) and the first part of the second stage (RoIAlign). The Mask R-CNN modules for object detection and segmentation are ignored. We use transfer learning in the following way. The Mask R-CNN is pre-trained on an object detection problem, and the segmentation model was pre-trained for detecting and encoding image regions on the MS COCO dataset [28].

The input of the model are images with varying sizes. Therefore, the images are resized with a scale that ensures that the smaller dimension is at least 800 and the longer dimension is maximum 1024 pixels. We apply padding to the scaled image to fix the image dimensions to 1024×1024.

The image is processed with the ResNet feature pyramid network of the ResNet-FPN convolutional backbone architecture for feature extraction of an entire image. This bottom-up approach extracts the features of the image with five stages of the ResNet [29] architecture, which has 101 layers. Each stage is incorporated into a top-down Feature Pyramid Network (FPN) network [30], which constructs higher resolution feature maps.

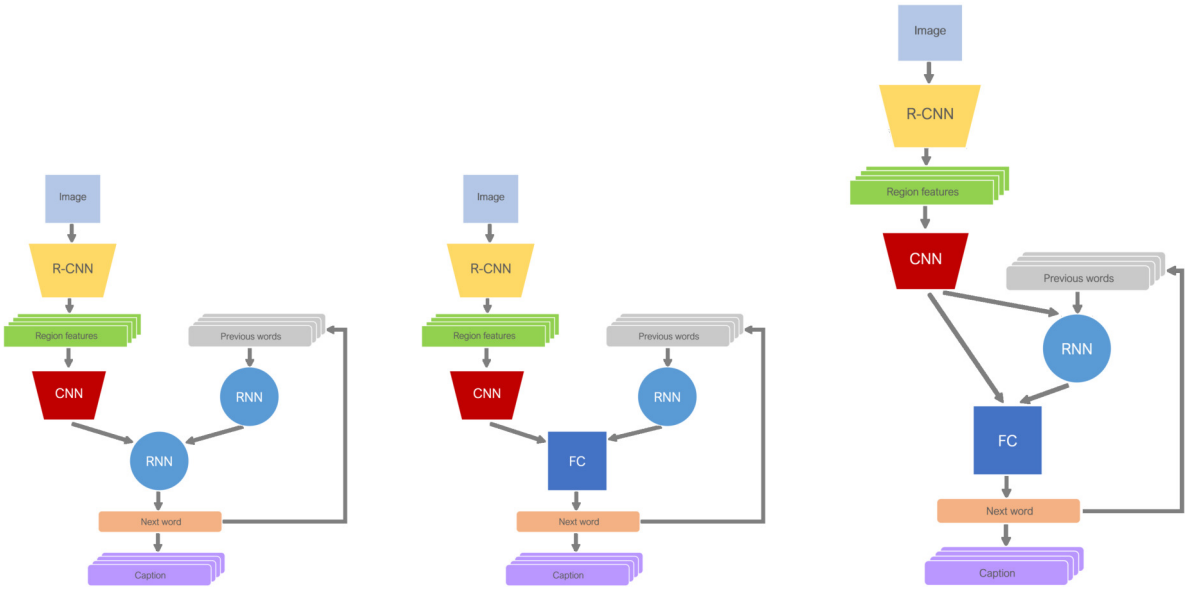
The proposed boxes, called anchors, are generated given a sliding window with proper scale and ratio. The RPN network ranks the anchors and chooses the ones that most likely contain objects. This process involves predicting foreground and background boxes from the anchors and their refinement. The output regions of the RPN are then processed with non-maximum suppression (NMS) to remove the highly overlapping regions. With an Intersection over Union (IoU) threshold of 0.7 of the NMS method, the region proposals are filtered according to their class probability of being positive (foreign) region. RoIs can be with different sizes, so the RoIAlign layer is proposed to generate small feature maps with size $7 \times 7 \times 256$ by applying bilinear interpolation. The outputs of the RoIAlign layer are the feature maps for every RoI.

C. Text generation deep learning architectures

The architectures of the proposed models are shown in Fig. 3. The application of recurrent neural networks (RNNs) in image captioning problems is discussed in [31]. An RNN can be used as either a decoder (generating words) or an encoder (encoding preceding words). In the proposed architectures, we utilize RNNs in both ways, as described below.

All three proposed architectures start similarly, by feeding the image through the R-CNN network that we reuse from the Mask R-CNN model (the yellow block named R-CNN in Fig. 3). This network creates features for each RoI of an image provided at the input. In Fig. 3, the FC blocks denote feed-forward networks which are represented by fully-connected dense networks, as described in the following text.

1) Inject Model (M1): Fig. 3a presents the diagram of our first model, M1. First, as mentioned earlier, the image is fed through the R-CNN network that we reuse from the Mask R-CNN model (the yellow block named R-CNN on



(a) Inject model - M1 (b) Merge model - M2 (c) Hybrid model - M3

Fig. 3: Diagrams of the proposed model architectures

Fig. 3a), thus generating features for each RoI of an image. A convolutional neural network (CNN) then processes these features and encodes each RoI. In parallel, a recurrent neural network (RNN) encodes the previous words (the blue circle in Fig. 3a). The features representing both the RoI of the image and the previous word embeddings (i.e., features) are fed into a decoder RNN (the second blue circle in Fig. 3a) that decodes the next word. Because the image features are injected into this RNN, we denote this model as the inject version. In our evaluations of this architecture, the decoder RNN is an LSTM with 256 units. The caption is created word by word in a loop with a predefined padding size.

2) Merge Model (M2): The second proposed architecture which we denote as the merge version or M2 is shown in Fig. 3b. It is identical to M1 in the way in which it creates features for the RoIs of the input images. However, unlike in the previous architecture, it uses a fully connected layer as a decoder. The features representing the image and the previous words are merged and passed to a fully connected (i.e., dense) layer, behaving as a decoder (instead of the second RNN used in M1), as shown by the blue FC block in Fig. 3b. The number of units in the fully connected layer is equal to the vocabulary size. Identical to the previous model, the caption is created word by word in a loop with a predefined padding size.

3) Hybrid Model (M3): Additionally, we propose a third architecture called a hybrid model (M3), which is shown in Fig. 3c. Leveraging the ideas from the former two models, the image features are concatenated with the word embeddings of the previously generated words and fed into the RNN network that encodes the previous context. The encoded context is concatenated with the image features, and two fully connected layers decode this vector representation into predicted word. The difference in the training between this model and the prior two is that this model is trained one-way, i.e., a caption by caption, unlike the multi-way training, a word by word, of the other methods.

In our experiments, the hybrid model (M3) uses two LSTM layers with 512 units for encoding the previous words, and two fully connected layers for generating captions: one fully connected layer with 1024 units and second fully connected layer with the number of units equivalent to the vocabulary size for the FC block in Fig. 3c.

D. Scoring metrics

Evaluating the output of a natural language generation model is a fundamentally difficult task. The most common way to assess the quality of automatically generated texts is a subjective evaluation by human experts [32]. However, human evaluation is not always attainable. Another approach is to use automatic evaluation metrics, such as METEOR [33] and BLEU [34], which were developed for machine translation. ROUGE [35], which was developed for text summarization, and CIDEr [36] and SPICE [37] which were developed for evaluating image captions. All these measures compute a score that indicates the similarity between the system output and one or more human-written reference texts.

E. Training details

We partitioned the dataset into three subsets of size $90,000$, $10,000$ and $8,077$ images for training, validation and testing,

respectively. Captions of the images in the validation and training subset were used for creating the vocabulary. All words are converted into lowercase. Words representing punctuation were removed. The final vocabulary has $36,413$ tokens.

Consequently, words are represented with one-hot encodings of size $36,413$. Each word is related to an integer that maps the word with its corresponding one-hot encoding. An embedding layer is used to encode the words into a representation with size 300. This layer's weights are frozen and initialized with weights from the GloVe (Global Vectors for Word Representation) [38] model, which is pre-trained on the Wikipedia corpus[2].

All models were trained with categorical cross-entropy loss function, Adam optimizer [39] with 0.001 learning rate and batch size 1024. Regions of the image are characterized by three-dimensional features of size $7 \times 7 \times 256$). For previous words, we use a frame with a padding size of 10, so for each word, we utilise the previous 10 words as features. If there are less than 10 previous words, features are padded with zeros to the required padding size.

All the models are implemented using the Python deep learning library Keras[3] with Tensorflow[4] backend. The training and testing were performed on NVIDIA Tesla K80 GPU on Windows Azure. Some experiments were also performed on an on-premises NVIDIA Titan V GPU.

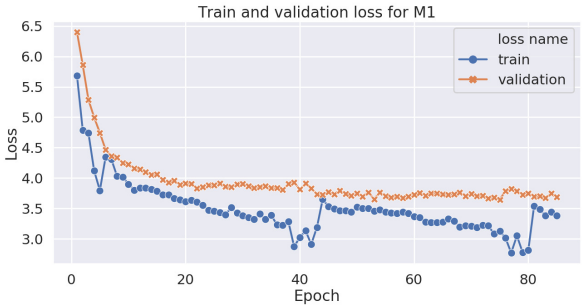
IV. Results and Discussion

We evaluated the three models using the evaluation metrics described in Section III-D. For each RoI of each image in the test set, all metrics were calculated and then averaged to get an average score the image.

The first two models, inject (M1) and merge (M2), were trained in 85 epochs using the ground truth regions of the images. Train and validation losses are shown in Fig. 4a (inject model - M1) and Fig. 4b (merge model - M2). Fig. 4a shows that for M1 in the first epochs both validation and training loss decrease. After about 40 epochs, the training loss starts oscillating between 3 and 4. Similarly, for model M2 the validation loss is decreasing and training loss is oscillating between 3 and 4, as shown in Fig. 4b. Fig. 4c shows oscillating training loss and decreasing validation loss for the hybrid model - M3.

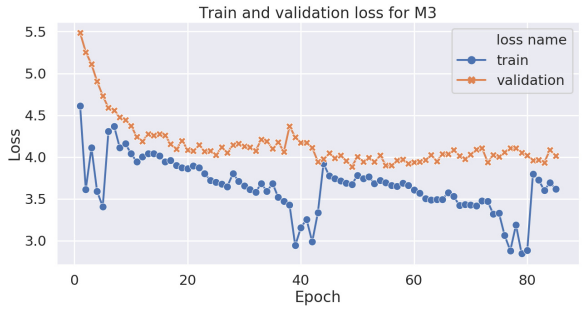
The average evaluation scores for each metric on the test set are shown in Table I. The table also includes information about the number of weights that need to be trained for each model.

The comparison of the inject (M1) and merge (M2) models highlights that the inject model has better performance. Although the merge model has comparable results, it achieves lower average scores for all metrics except BLEU-1. This contradicts the findings of [31], which showed that the merge version generally outperforms the inject version of models. We could hypothesize that the RNN decoder outperforms the



(a) Model M1

(b) Model M2



(c) Model M3

Fig. 4: Train and validation loss of (a) the inject model (M1), (b) the merge model (M2) and (c) the hybrid model (M3)

fully connected decoder, as opposed to the findings of [31] which demonstrate that applying fully connected layer as a decoder leads to better performance. However, their task differs from ours since we predict multiple captions for an image as opposed to predicting a single caption. Moreover, both problems require different dataset types, that is "image - single caption pairs" for single caption generation and "image - multiple caption pairs" for multiple caption generation. Therefore, we cannot precisely determine if one architecture is better than another.

Even though the hybrid model (M3) was trained differently than models M1 and M2, it was evaluated with the same test set. The evaluation shows that the M3 model achieves lower scores. We could hypothesize that the reason for the low predictive performance of this model could be the fact

[2]https://nlp.stanford.edu/projects/glove/, last visited: 22.05.2020
[3]https://keras.io/, last visited: 22.05.2020
[4]https://www.tensorflow.org/, last visited: 22.05.2020

TABLE I: Evaluation results for the proposed models M1 (inject model), M2 (merge model) and M3 (hybrid model).

Model	#Weights	SPICE	ROUGE-L	METEOR	CIDEr	BLEU-1	BLEU-2	BLEU-3	BLEU-4
M1	17M	**0.1387**	**0.3593**	**0.1411**	**0.9935**	0.3073	**0.1600**	**0.0946**	**0.0526**
M2	80M	0.1274	0.3476	0.1349	0.9142	**0.3136**	**0.1600**	0.0941	0.0521
M3	32M	0.0452	0.1643	0.0524	0.2553	0.1561	0.0664	0.0386	0.0277

that it was trained a caption by a caption and could benefit from more training.

BLEU [34] measures how close a candidate sequence is to a reference sequence, more concisely, the hits of n-grams of a candidate sequence to the reference. According to the results, we could hypothesize that M2 performs better in terms of matching smaller n-grams, that is, unigrams and possibly bigrams. However, for matching longer n-grams, M1 achieves better results. This is confirmed with ROUGE-L [35], which applies the concept of the Longest Common Subsequence (LCS). The value of this metric is higher for M1.

CIDEr [36] measures how often n-grams in the candidate sentence are present in the reference sentences, while METEOR [33] is based on the harmonic mean of unigram precision and recall, where recall is weighted higher. Both metrics map the words in their stem or root forms. M1 shows better performance for both metrics. Therefore we can infer that this model generates words that perhaps may not be the exact match of the reference words, but they nevertheless have the same root form.

SPICE [37] measures how effectively image captions recover objects, attributes and the relations between them. It is based on the agreement of the scene-graph tuples of the candidate sentence and all reference sentences. The M1 model, again, achieves the best performance leading to the conclusion that this model effectively describes the image scene.

The number of trainable weights of the models is included in Table I. The M1 model is the smallest model out of the three models. That could be the reason for the best performance of this model, which is learning fewer weights given the same training time. Having a bigger model with many trainable weights has been the preferred way for learning more complex relationships in images. However, larger models also require more training time for learning all the weights. Therefore, with the results from these experiments, we can infer that the smaller model is more practical and has the best performance in this setting. Also, regarding the weaker performance of the M3 model, we can conclude that the multi-way training (word by word) is preferred over the more difficult process of one-way training (caption by caption).

A. Extensive analysis of the capability of the models

Evaluating the models based on the n-gram evaluation metrics limits us to understand the relative strengths and weaknesses of the models. Therefore, we use the property of the SPICE metrics that enables us to divide the metric value into meaningful categories.

In Table II, we review the performance of the models from different aspects. The table contains F-scores for the subcategories from which SPICE is calculated, that is objects, their attributes, and relations between them. The M1 model surpasses the other models for all of the categories, except the size category, where the M2 model is finer. This effect means that the M2 model caption generator is better for capturing the size of the objects than the other models. The M3 model is inferior in these settings, and we can see that the crucial shortcoming of the model is the cardinality, so the model is not able to count while generating the captions.

From the evaluation results, we can conclude that the models perform well at capturing objects present at the image and their cardinality. However, they fail to describe the attributes of the objects. We could hypothesize that such behaviour is expected since the part of the models that extracts image features is pre-trained on an object detection task and therefore could potentially be biased towards detecting objects rather than describing them in details. Therefore, because the text generation models are separated from the CNN model for creating the image features, one way of improving is to refine the features of the CNN model by additionally training the model on attribute prediction, not solely on object detection. In this way, the image features should be expected to include more information about the attributes of the object and hence help the text generation models to create better captions.

B. Qualitative results

We present example predicted captions for ground truth regions from models M1 (inject) and M2 (merge) in Fig. 5. Predictions from M1 are shown on the left, while predictions from M2 in the right. For brevity, we plot only one region caption per image. For each model, one good, one quite good and one not good example are displayed.

The first row presents captions classified as good. The predictions are made with padding size 10, i.e. each generated caption has length 10. However, from the examples, we can infer that for some regions, this padding size is too big. Both models generate descriptive captions with specific length and fill the rest with words unrelated to the image. Nevertheless, we classify such captions as good. Quite good captions are those related to the image with minor errors (second row). For example, M2 generates the following caption "child wearing a blue shirt". As we can see, the child is wearing a white shirt. We can conclude that even though the colour is incorrect, the main context of the region is described. The last row presents captions classified as not good. These captions are unrelated to the region, which they describe.

TABLE II: F-scores from SPICE by semantic proposition subcategory. The models models M1 (inject model), M2 (merge model) and M3 (hybrid model) are compared with the SPICE metric for object, relation, attribute, color, cardinality and size.

Model	SPICE	Relation	Cardinality	Attribute	Size	Color	Object
M1	**0.1387**	**0.0595**	**0.1241**	**0.0618**	0.0493	**0.0627**	**0.1989**
M2	0.1274	0.0456	0.1151	0.0548	**0.0502**	0.0499	0.1854
M3	0.0452	0.0197	0.0000	0.0219	0.0066	0.0408	0.0671

Fig. 5: Examples of generated region captions for the inject model, M1, (left) and the merge model, M2 (right)

V. Conclusions

This paper investigated the problem of automatically generating descriptions for RoIs in images. The aim is to investigate the appropriate model for generating text that describes RoI in images. The Mask R-CNN model trained for image classification was modified and used for RoI feature extraction. For caption generation, three model versions were proposed. In the first version, called inject model, image features are injected into a decoder RNN. In the second version, called merge model, image features and previous words features are concatenated and fed into a fully connected layer as a decoder. The third version, called hybrid model, the image features are fed into a decoder RNN but the caption is generated one-way instead of generating word by word as in the previous two models.

We evaluated the proposed models with several text evaluation metrics. The results show that the models M1 (inject model) and M2 (merge model) are better than M3 (hybrid model), with M1 having the best performance. Also, the M1 model has the smallest number of trainable weights out of the three models and still is the best performing model. The experimental results demonstrate that injecting image features into a decoder RNN while generating a caption word by word is the best performing architecture among the architectures explored in this paper. The extended evaluation represents the shortcomings of the models to describe the attributes of the objects in the images. Hence, future experiments should examine the models after training the CNN feature extractor on attribute prediction.

The visual text generation models are impressive in most of the cases, but they also have faults. Show-and-Fool [40] is a model created for attacking image captioning models with adversarial perturbations in machine vision and perception to produce randomly chosen captions that are not relevant to the image. Therefore, the future work could focus on applying such attacking model for evaluating the robustness of the proposed model. An alternative implementation of this model is to build a more robust image captioning model using an attack model into a GAN network. Likewise, for caption generation it could be interesting to investigate the capability of networks consisted of attention only, such as the Transformer [19] approach, to encode and decode both the

context of the image and the text.

ACKNOWLEDGMENT

This work was partially financed by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University, Skopje, North Macedonia. We also gratefully acknowledge the support of Microsoft Azure for Research and the NVIDIA Corporation through grants providing GPU resources for this work.

REFERENCES

[1] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 684–699.

[2] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3588–3597.

[3] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 135–11 145.

[4] M. 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, p. 118, 2019.

[5] W. Hechun and Z. Xiaohong, "Survey of deep learning based object detection," in *Proceedings of the 2nd International Conference on Big Data Technologies*, 2019, pp. 149–153.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014.

[11] 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.

[12] 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.

[13] 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.

[14] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.

[15] Z. Yang, Y.-J. Zhang, S. ur Rehman, and Y. Huang, "Image captioning with object detection and localization," in *International Conference on Image and Graphics*. Springer, 2017, pp. 109–118.

[16] X. Zhu, L. Li, J. Liu, H. Peng, and X. Niu, "Captioning transformer with stacked attention modules," *Applied Sciences*, vol. 8, no. 5, p. 739, 2018.

[17] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[18] S. He, W. Liao, H. R. Tavakoli, M. Yang, B. Rosenhahn, and N. Pugeault, "Image captioning through image transformer," *CoRR*, 2020.

[19] 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.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[21] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4565–4574.

[22] L. Yang, K. Tang, J. Yang, and L.-J. Li, "Dense captioning with joint inference and visual context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2193–2202.

[23] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[24] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, and J. Shao, "Context and attribute grounded dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6241–6250.

[25] X. Li, S. Jiang, and J. Han, "Learning object context for dense captioning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8650–8657.

[26] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2407–2415.

[27] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[31] M. Tanti, A. Gatt, and K. P. Camilleri, "What is the role of recurrent neural networks (rnns) in an image caption generator?" in *The 10th International Natural Language Generation conference*, vol. abs/1708.02043, 2017, p. 51. [Online]. Available: http://arxiv.org/abs/1708.02043

[32] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, "Automatic description generation from images: A survey of models, datasets, and evaluation measures," *Journal of Artificial Intelligence Research*, vol. 55, pp. 409–442, 2016.

[33] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 376–380.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[35] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[36] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[37] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *European Conference on Computer Vision*. Springer, 2016, pp. 382–398.

[38] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[40] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, "Attacking visual language grounding with adversarial examples: A case study on neural image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2587–2597. [Online]. Available: https://www.aclweb.org/anthology/P18-1241