SeqCapsGAN: Generating Stylized Image Captions using Capsule Generative Adversarial Network

Ali Bibi
School of Computer Science & Eng.
Nanyang Technological Univ., Singapore
ali.bibi@outlook.com

Helmi Abidi
School of Computer Science & Eng.
Nanyang Technological Univ., Singapore
helmi.abidi@gmx.de

Oussema Dhaouadi
School of Computer Science & Eng.
Nanyang Technological Univ., Singapore
oussema.dhaouadi@gmail.com

***Abstract*—Since the dawn of natural language processing, image captioning has been the focus of many research groups. It gained major importance because of its numerous applications. Describing the content of a visual input accurately can add relevant features to current social networks. In addition, it is able to help blind people understand the environment by reading the generated caption to them. Most of these applications require human-like captions, which include adjectives reflecting positive or negative sentiments. This research specializes in stylized image captioning. In this work a new network architecture called SeqCapsGAN was implemented based on state-of-the-art techniques. Both Generative Adversarial Networks (GANs) and CapsNet have shown good results in generating and classifying sentences. Thus these techniques were combined in this paper and SeqCapsGAN has outperformed many of the studied baseline models in discussed evaluation metrics.**

***Index Terms*—Generative Adversarial Network (GAN), Capsule Network (CapsNet), Natural Language Processing (NLP), Stylized Image Captioning**

I. Introduction

Deep Learning knows a rapid progress since many years. This is recognizable by the number of emerging architectures, training techniques and the new applications it opened up. Similar to other fields, different NLP tasks are now made possible. Results of different problems, that were a few years ago considered extremely complicated with classical methods, are now very promising. Among these tasks is image captioning. It consists in automatically generating a description to an input image. Success in this task can be very helpful in different social applications. They can enhance the engagement level of users through chat-bots [1] or assess people on social networks by captioning their content [2]. This research field started gaining a progressively increasing attention after the successes made by neural networks in computer vision, where object detection is achieved with a high accuracy and by seq2seq models. These enable generating coherent texts easily. In addition to that, large image-caption data-sets are now publicly available such as MSCOCO [3] and Senticap data-set [4] used in this work. Building on all of this, current image-captioning models have shown considerable performance.

Nevertheless, these models are able to describe images only in a factual way, which is similar to enumerating existing objects in the image and annotating how they are connected. This description does not include subjective adjectives like ”nice” or ”bad” as it disregards human sentiments. This makes the generated captions unnatural and different from the way humans interpret visual content. Humans prefer stylistic captions that reflect their emotions in a series of adjectives. This confirmed by the fact that majority of existing online captions are of this kind [4]. The difference between factual and stylized captions (whether positive or negative) is exhibited by the Fig. 1 below:

Fig. 1: Difference between neutral, positive and negative captions

A generated caption is considered successful when it satisfies a series of criteria. First of all, it has to reflect the content of the image. This means relevant objects in the image and their relationships should be indicated. Second, appropriate adjectives have to be used and in a meaningful way, which means the sentence is fluent and grammatically correct. So a successful stylized captioning model should be able to generate captions achieving these criteria whether the intention is to describe the image in a positive or negative manner. Furthermore generated captions should have different styles, i.e. a variety of adjectives and grammatical structures.

Different models tried to describe images subjectively but they did not achieve a good performance due to multiple reasons. Most notably stylistic data-sets are unfortunately small, so that the network is not able to learn a variety of adjectives and stylistic patterns. We can illustrate the problem by this caption ”a dead man doing a clever trick on a skateboard at a skate park” from the system of Mathews *et al* [4]. The man is not dead but ”dead” was used because it is among the most frequent negative adjectives. This research work will try to surpass mentioned challenges.

II. Related Work

The proposed solution in this work is mainly related to two research topics: image captioning and Generative Adversarial Networks (GANs). In this section, we provide the background for both topics.

Image Captioning

Early image captioning methods relied on a retrieval-based algorithm [5] [6] [7]. They depend on an image database. The input image is compared with existing images in the database. Similarity checks are performed and similar images are extracted. A language model uses the captions of retrieved images in order to generate a new description that fits the input image. The downside of this approach is that it relies heavily on the database.

Recent approaches adopted an encoder-decoder framework of Vinyals *et al* [8], which is the basis of modern image captioning systems. The first step of this framework is to encode the visual content and generate visual features. This is performed by a Convolutional Neural Network (CNN). Then comes the role of the decoder, which will interpret the generated features and then generate the caption. Many improved approaches [9] [10] [11] are developed on this framework. The differences between them is often related to the architecture of recurrent neural network. Newly developed ones use a Long-Short Term Memory (LSTM) network as a decoder.

Adding an attention model to the framework discussed above has shown a considerable performance. Many models [12] [13] [14] gained recent success, which motivated researchers to focus on the embedding of the attention component in the encoder or decoder. In fact it is possible to apply visual or language attention models [15] [16] [17]. First approaches [16] [18] applied a top-down visual attention model. The model assigned weights to image regions which enabled a deeper understanding of the visual content. A step further came later by combining bottom-up and top-down attention models [19]. In comparison to visual attention models, semantic attention attending to visual features was applied by You *et al* [17]. - stylistic image captioning.

All models mentioned above tried to describe images in an objective, factual way. But currently stylistic captions are popular thanks to two recent models: StyleNet [2] and Senticap [4]. They described images with specific styles and sentiments respectively.

In this work we want to allow image captioning with both specific styles and sentiments.

Generative Adversarial Network (GAN)

Ian Goodfellow proposes the idea of Generative Adversarial Networks [20]. It was applied first of all in the Computer Vision area. It is s a probabilistic model that consists of a generator and a discriminator. The purpose of the generator is to map latent vectors to images that resemble the data distribution closely and to try to fool the discriminator into misclassifying fake (i.e. sample drawn by the generator) or real images (i.e. sample coming from the ground truth data).

GANs are designed to work on a continuous data space [20]. But in the context of caption generation, we are dealing with a discrete data distribution. To handle this, the problem of sentence generation is formulated as a reinforcement learning problem [21]. For each generated word, the discriminator calculates a reward and provides it to the generator. Based on this reward the generator calculates the gradients and updates its parameters. This technique was applied by recent models [22] [23] to generate sentiment-bearing text.

III. Methodology

A. Framework

Fig. 2 shows the framework of the proposed SeqCapsGAN for stylized image captioning. The framework is divided into

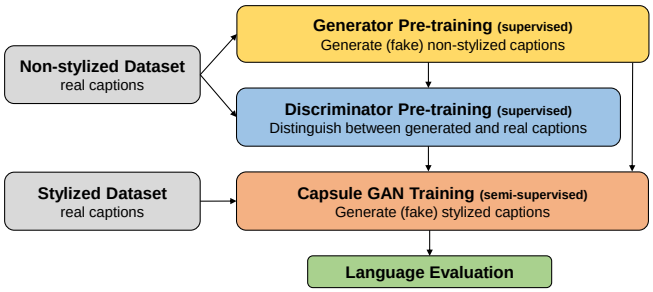

Fig. 2: Framework of SeqCapsGAN for stylized image captioning.

three modules. In the generator pre-training, neutral image captions are generated using an LSTM network with attention mechanism. The network learns the mapping between features extracted from the images and the captions. By using non-stylized data-set, the generator learns the general structure of the language and focuses on understanding the morphology and how to built a coherent sentence. The discriminator pre-training prepares this classifier to distinguish between captions generated by the generator (fake captions) and captions from the data-set (real captions). This step is important for the adversarial training in the GAN framework, since differentiating between sentences stemming originally from the same data-set is not straightforward. It prepares both components of the GAN for a competitive training in a fair manner. The third module runs the GAN engine by feeding its generator with images and its discriminator with stylized captions. The generator tries to learn adjectives and adverbs indirectly based on the feedback of the discriminator in identifying real and fake captions. Hence, the main focus of the generator is to figure out how to incorporate sentiments into a sentence. Finally, an evaluation of the generated captions is performed for comparison purposes. The modules are described in the next subsections.

B. Generator

The generator consists of three components: a feature extraction module, an encoder and a decoder. Fig. 4 shows the main components generator. The generator inputs an image and a desired sentiment code, and outputs a coherent stylized sentence pertinent to the inputs. This network operates in three modes, namely in the pre-training mode, where the

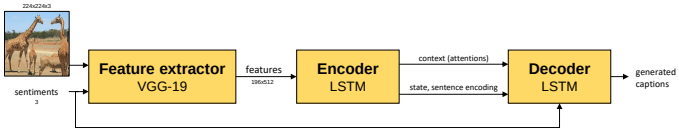

Fig. 3: Generator's components.

network tries to learn the mapping between visual features and linguistic description, in an adversarial training mode in which the network learns the sentiments in order to generate stylized sentences based on a reward earned from a discriminator, and in a sampling mode in which the generator simply infers a description from the image according to the input sentiment.

1) Feature extractor:: To extract visual features from an image, the pre-trained VGG-19 [24] is used. The VGG-19 network is a convolutional neutral network that has 19 layers and is able to classify images into 1000 classes. We access one of the last layers to represent an image in a feature map of size 196x512. Since the top-5 error is 9.12%, this network should be sufficient to describe an image in 100352 scalar values. The features are fed to the encoder.

2) LSTM Encoder and Decoder:: The LSTM encoder learns to describe the context of an image by the means of the extracted features. Visual features are fed to a state and memory initializer, where the initial state and memory are learned. These are fed to LSTM blocks in sequence in order to learn the sequence dependency. The LSTM entities also input a context vector provided by attention blocks, where the most attractive features are highlighted depending on the current state of the LSTM block. The attention vector and the output of the LSTM are fed into a decoder, where the decoded outputs are the probability of every single word in the vocabulary. The index of the most probable word is given as input to the next LSTM block, where the learning in sequence takes place. In the learning mode, we feed the block with the ground truth captions instead of the previously predicted word. This is known as the teaching mode, which guides the LSTM block generating words similar to the ground truth. The structure of the generator's encoder and decoder in sampling mode is shown in figure 4.

a) State and Memory Initializer: To initialize the LSTM state and memory, the visual features are fully connected to the state neurons and memory neurons. The weights of the networks are learned through backpropagation.

b) Attention Mechanism: An attention block inputs the visual feature vector, the LSTM state h and outputs a vector representing the context of the image. As a first step, the features are projected into some space. The projection is represented by a fully connected layer, where the dimension of the input is the same as the output. The state h is projected in the feature space using a dense layer. The projected features and the transformed state are added and multiplied by an attention layer. The result is activated by a softmax function and stored in a vector α. Every scalar value of the attention vector α is associated to a value of the feature vector, indicating the relevance and importance of the feature in building a sentence. The features are then multiplied by the attentions α to form the weighted feature vector. This vector is further multiplied by an intensity vector β, which indicates relevant part of the features to create the current word. The result is the context at current time step t.

c) LSTM Cell: The context feature at time t is fed to the LSTM block as an input. The sentiment input vector indicating whether the generated sentence should be sentimental positive or negative and the embedding of the previous decoded word are concatenated to the input vector. The state h and the memory c are propagated to the next LSTMs until the end word is generated or until the the maximum number of words is reached.

d) Decoder: The decoder sums up the output of the LSTM, the projected context vector, the projected state of LSTM, the sentiment vector input and the previously generated word or ground truth caption depending on the operating mode of the generator. The result is activated by the $\tanh$ function and then projected onto the vocabulary space. The index of the most probable word is the decoded value.

C. Discriminator

CNNs are known by their high performance in recognition tasks. However, they are suffering of lack of information dropped out by the pooling layers. In fact, this operation losses the global pose information of a feature, i.e. the position and the relationship between features themselves. The Capsule Netowrk proposed by Sabour et al. [25] tackles this problem by changing the basic of a neural network, namely the neurons by an entity called capsule, which routes information in a sophisticated fashion without loosing much in information.

1) Drawbacks of CNNs: Capsule Networks are the state-of-the-art classifiers in computer vision field which operate on images. Since our structure requires the operation of 1D signal, namely a sentence, we introduce the 1D Capsule Network. This network has basically the same structure as in the original paper but adjusts capsules to 1D input tensors and changes the convolutional 2D operations to 1D. The main purpose of capsules is the encapsulation of relevant information about the features they are recognizing. They not only learn the recognition in the sense of existence but expands it to the sense of how the features are present. The characteristics of the features also called instantiation parameters are stored in the capsule in form of a vector and the presence of that feature is represented by the length of the vector in some n-D space. The main problem of CNN is that the classification is solely based on the existence of the features without considering the relationship between them, which is lost after performing the pooling operation. For instance, a CNN could not distinguish between the sentences "Jack gave Oliver an advice." and "Oliver an advice gave Jack.", since they represent the same features. They would be classified to a same class. However these sentences are not the same when using the Capsule Network. In fact, the direction of the feature vectors in some space represented by the capsules are not the same but their

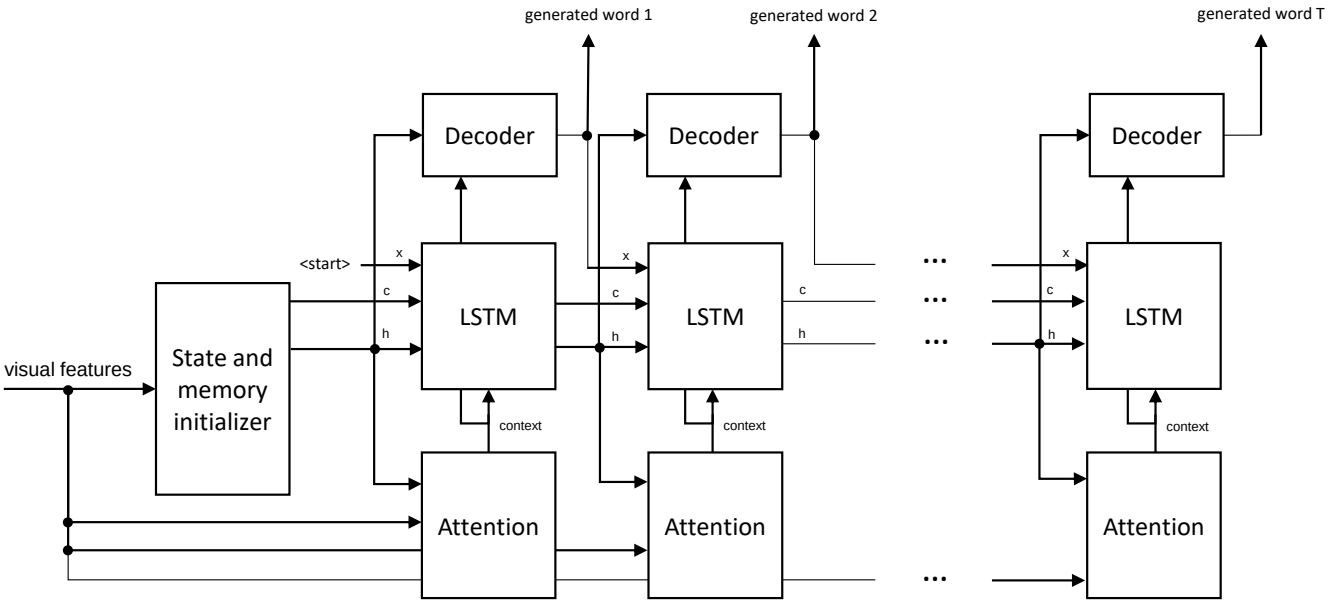

Fig. 4: Generator Encoder and Decoder in Sampling Mode

lengths are the same. This illustrates the relevance of encoding the relationship between features used for the prediction.

2) Capsule Structure: A capsule entity is similar to a neuron but operates on vectors instead of scalar values. It is represented by a vector that encapsulates feature information. The computation inside it described as follows:

a) Matrix Multiplication: A capsule i represented by a vector $\mathbf{u}_i$ in the layer L stores the instantiation parameters as entries of the vector. The existence probability is encoded in its length $\mathbf{u}_i$ A multiplication by a learnable $\mathbf{W}_{j,i}$ is performed, where transformations are learned. This allows the capsule to learn the relationship between lower level capsule i and high level capsule j. The output $\hat{\mathbf{u}}_{j|i}$ represents where the higher level feature should be compared to the detected pose of the lower level feature.

b) Scalar Weighting: The predictions $\hat{\mathbf{u}}_{j|i}$ are scaled with $c_{j,i}$, which are computed by the dynamic routing algorithm. If the result is in some space close to a higher level capsule v_j, the latter will be activated. The higher level capsule is formed by the lower level feature entities that had agreed on the same capsule.

c) Prediction Summation: This operation is similar to the neuron entity, where a weighted summation is performed. In this case, the summation is applied on vectors.

d) Capsule Activation: The squash function defined in [25] is applied to compute the capsule v_j:

$$\mathbf{v}_j = squash(\mathbf{s}_j) = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}. \tag{1}$$

This function ensures the length of the vector to be smaller than 1 in order to present the existence probability on its length without changing its direction.

3) Dynamic Routing: The dynamic routing algorithm is the engine of routing between two capsule layers. It results in computing the scaling factors $c_{j,i}$ and the higher level capsule. Its is run over a specific number of iterations, typically 3 iterations. A higher number of iterations could lead to overfitting. The algorithm is described in Algorithm 1.

Algorithm 1 Dynamic routing algorithm [25]

procedure DYNAMIC ROUTING($\hat{\mathbf{u}}_{j|i}$, r, Ω_L)
  $\forall\, i \in \Omega_L$ and $\forall\, j \in \Omega_{L+1} : b_{j,i} \leftarrow 0$.
  for r iterations **do**
    $\forall\, i \in \Omega_L : c_{j,i} \leftarrow softmax(\mathbf{b}_i)$
    $\forall\, j \in \Omega_{L+1} : \mathbf{s}_j \leftarrow \sum_i c_{j,i}\hat{\mathbf{u}}_{j|i}$
    $\forall\, j \in \Omega_{L+1} : \mathbf{v}_j \leftarrow squash(\mathbf{s}_j)$
    $\forall\, i \in \Omega_L$ and $\forall\, j \in \Omega_{L+1} : b_{j,i} \leftarrow b_{j,i} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$
  return $\mathbf{v}_j$

4) 1D Capsule Network: As mentioned, the capsule network focuses on encoding the relationship between features. To encourage learning robust codes, the capsule network introduces a decoder using a fully connected network that has a low penalty on the total loss function.

The structure of the capsule network starts by extracting features from the input vector using a convolutional layer. The resulting feature space is split into sub-spaces to form the capsule entities of primary capsule layer. The next layer is the digit capsule layer, formed using the dynamic routing algorithm. The size of this layer is equal to the number of classes. Hence, each capsule from the digital capsule layer, called parent capsule, presents a class.

The so-called margin loss is the optimization function used to

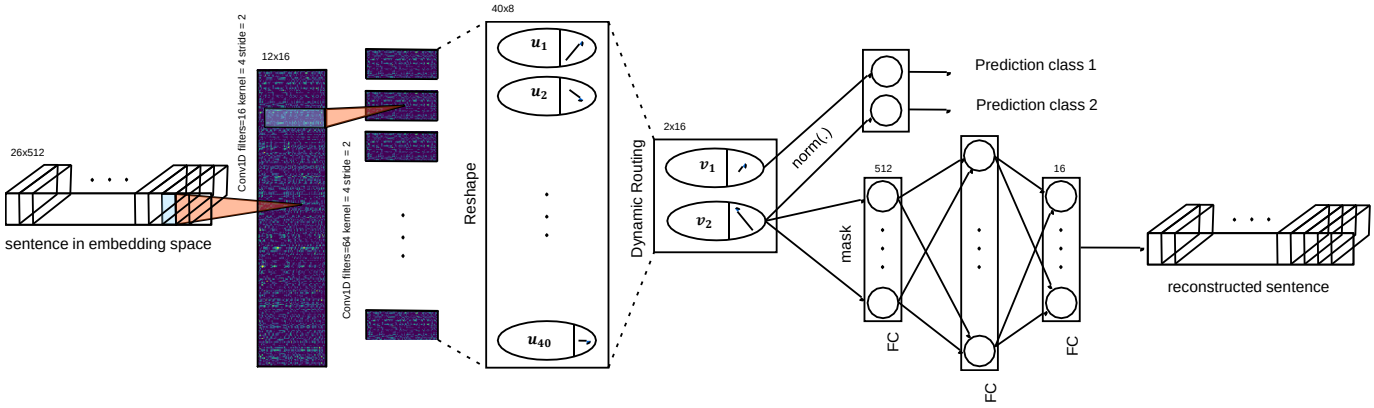

Fig. 5: 1D Capsule Network applied on sentences

train the capsule network [25]:

$$\mathcal{L}_{margin} = \sum_{i \in \Omega_{last}} T_i \max(0, m^+ - \|\mathbf{v}_i\|)^2 + \lambda(1 - T_i) \max(0, \|\mathbf{v}_i\| - m^-)^2, \tag{2}$$

where T_i, m^- and m^+ are hyperparameters. The scalar value λ is set to 0.5 for numerical stability. In addition to the margin loss, a reconstruction penalty is added to the overall loss. This forces the network to learn significant capsule instantiation parameters from which the network is able to reconstruct the input. This loss function is simply the Mean Squared Error (MSE) defined by the Euclidian distance between the original input and the reconstruction. The overall function is defined as follows:

$$\mathcal{L} = \gamma \cdot \mathcal{L}_{margin} + (1 - \gamma) \cdot \mathcal{L}_{reconst}$$
$$\mathcal{L}_{reconst} = \sum_i (\mathbf{x}_i - \mathbf{x}_{i,reconst})^2, \tag{3}$$

where $\gamma = 0.9$. Figure 5 presents the overall structure of 1D-CapsNet using dynamic routing.

D. SeqCapsGAN

This section presents a powerful tool for generating a complex and high dimensional distribution without learning the likelihood of a word explicitly from the ground truth data, i.e. the backpropagation loss is not computed at the output of the generator but rather at the output of another network called a discriminator.

Ian Goodfellow et al. proposed this idea to avoid intractability of distribution in order to compute the likelihood and presented in 2014 the Generative Adversarial Network (GAN) [26] as a probabilistic model that consists of a generator and a discriminator. (Figure 6) We adjust the original framework that operates on images to a natural language processing framework. The purpose of the generator is to map input features to sentences that resemble the data distribution closely (the ground truth captions) and to try to fool the discriminator into classifying fake (i.e. sample drawn by the generator) or real captions (i.e. sample coming from the ground truth data). The idea was inspired by two-player zero-sum game [27]. The generator is trying to maximize the chance of the discriminator misclassifying the generated caption and the discriminator is, in turn, trying to maximize its chance of correctly distinguishing the incoming generated caption from the ground truth data.

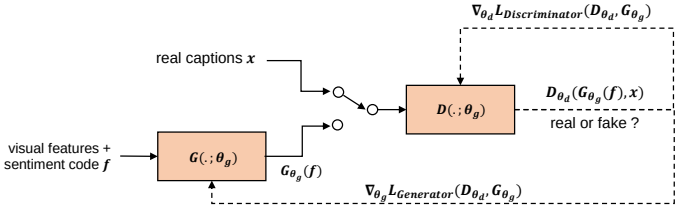

Fig. 6: Generative Adversarial Network.

Algorithm 2 SeqCapsGAN adversarial training algorithm

for r iterations **do**
  for k_d steps **do**
    Randomly select a minibatch of m images and extract the corresponding VGG-19 features and sentiments $\{\mathbf{f}^{(1)}, ..., \mathbf{f}^{(m)}\}$
    Sample minibatch of m examples $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\}$ from the ground truth data
    Update the discriminator by descending its stochastic gradient using equation 8
  end for
  for k_g steps **do**
    Randomly select a minibatch of m images and extract the corresponding VGG-19 features $\{\mathbf{f}^{(1)}, ..., \mathbf{f}^{(m)}\}$
    Update the generator by descending its stochastic gradient using equation 4
  end for
end for

The proposed GAN differs from the conventional GAN in

TABLE I: BLEU, ROUGE and CIDEr scores for the CNN and CAPS based GAN models

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr
CNN+RNN	48.15	27.80	16.65	10.25	36.35	55.10
ANP-Replace	48.15	28.30	17.05	10.50	36.45	55.85
ANP-Scoring	48.10	28.30	17.15	10.60	36.35	56.25
RNN-Transfer	48.55	29.25	18.30	11.50	36.95	55.00
SentiCap	49.55	30.15	18.90	11.95	37.20	58.10
SF-LSTM + Adap	50.40	**30.90**	**19.60**	**12.70**	38.00	**59.85**
SeqCNNGAN	47.53	24.49	13.01	7.19	37.60	31.34
SeqCapsGAN	**52.34**	29.94	17.09	10.17	**41.17**	45.40

(a) **SeqCNNGAN:**
P: a cow stands on a sunny beach with its wings extended .
N: a black cows standing in the dead grass of a pasture .
SeqCapsGAN:
P: a cow standing in a field next to a beautiful tree .
N: black cows are standing on a rough hill in a pasture .

(b) **SeqCNNGAN:**
P: a elephant wears a beautiful piece of patterned fabric .
N: the elephant is wearing a dead piece of cloth .
SeqCapsGAN:
P: an elephant wears a beautiful piece of patterned fabric .
N: an elephant with an ugly head and a stupid outfit .

(c) **SeqCNNGAN:**
P: a bus driving down a beautiful street in a nice city .
N: a bus driving down a lonely road on a lonely road .
SeqCapsGAN:
P: a large bus drives through a wide city great street .
N: a green bus drives down a lonely street in a lonely city .

(d) **SeqCNNGAN:**
P: a woman holding an umbrella covering themselves from good rain .
N: a woman in a girl with a big umbrella walking in the rain .
SeqCapsGAN:
P: a happy child sitting on a bench with a beautiful orange .
N: a woman with a crazy hat is sitting on a bench with a little girl in front of her .

(e) **SeqCNNGAN:**
P: a clock tower with a beautiful structure in the background .
N: a clock tower with a damaged building in the background .
SeqCapsGAN:
a clock tower with a clock stretches into the sky on a nice day .
N: a clock tower with a ugly building in the background .

Fig. 7: Image captions generated by SeqCNNGAN and SeqCapsGAN

terms of update policies. In fact, the generator minimizes the following two objective functions depending on its operation mode:

$$L_G(\theta_G) = \lambda L_{G,adv}(\theta_G) + L_{G,gen}(\theta_G), \tag{4}$$

where $L_{G,adv}(\theta_G)$ is the loss regarding the adversarial training (maximizing the penalty of the discriminator), $L_{G,gen}(\theta_G)$ is the loss regarding the generation of captions and λ is a pre-defined parameter. In the pre-training mode the generator solely minimizes the following function:

$$L_{G,gen}(\theta_G) = -\sum_{1\leq t\leq T} \log\left(p_w\left(x_t|\hat{a}_t, h_t\right)\right) + \tag{5}$$

$$\gamma \sum_{1\leq k\leq K}\left(1 - \sum_{1\leq t\leq T} a_{tk}\right)^2, \tag{6}$$

where p_w are the outputs of the LSTM blocks, γ is a regulation parameter and $\mathbf{a}_t \in \mathbb{R}^K$ is the attention vector of the generated word at time step t (K is the size of the visual feature vector). In the adversarial training mode, the generator optimizes both the generation loss $L_{G,gen}(\theta_G)$ and the adversarial loss $L_{G,adv}(\theta_G)$, which has the following equation:

$$L_{G,adv}(\theta_G) = \sum_{1\leq t\leq T} G_{\theta_G}\left(x_t|x_{1:t-1}, \hat{a}_t\right) \cdot Z_{D_{\theta_D}}^{G_{\theta_G}}\left(x_{1:t}\right), \tag{7}$$

where $G_{\theta_G}\left(x_t|x_{1:t-1}, \hat{a}_t\right)$ is the generated word given the previous words acquired from the LSTM output and $Z_{D_{\theta_D}}^{G_\theta}\left(x_{1:t}\right)$ is the reward value provided by the discriminator to the generator [28].

The discriminator optimizes the following loss function according the WGAN settings [29]:

$$\max_{w\in\mathcal{W}} E_{\mathbf{x}\in GTdata}\left[f_w(\mathbf{x};\theta_d)\right] - E_{\mathbf{f}\in feat.set}\left[f_w(G_{\theta_g}(\mathbf{z});\theta_d)\right], \tag{8}$$

where $\{f_w\}_{w\in\mathcal{W}}$ is a parametrised family of K-Lipschitz functions for some constant K (such as the sigmoid, tanh, elu, softplus, etc).[1]

After pre-training the LSTM-based generator and the Capsule-based discriminator, both networks are merged to-

[1]A function f: $\mathcal{A} \to \mathbb{R}^m$, is said to be L-Lipshitz[2], $K \geq 0$, if the constraint $|f(a) - f(b)| \leq |a - b|$ is fulfilled for all $a, b \in \mathcal{A}$ [30]

gether to perform the adversarial training. They following the update policies described in the algorithm 2.

If the discriminator learns slowly and cannot distinguish between the samples after one learning step, it is necessary to train it k_d-times before optimizing the generator. Likewise, the generator may be weaker than the discriminator. Training it k_g-times gives the generator the chance to improve itself.

The implementation of this model is uploaded to github [1].

IV. Experiments

A. Datasets

We pre-train the generator and the discriminator using the Microsoft COCO data-set to help the networks learn the mapping between visual features and the morphology of the language. We further train the generator in the GAN framework using the SentiCap data-set to learn the including sentiments into the captions.

- *Microsoft COCO Dataset* [3]: It is an image-caption data-sets. It includes 82K+ images and 413k+ captions. This data-set was used to pre-train the generator and discriminator as it has neutral captions.
- *SentiCap Dataset* [4]: This data-set has sentiment-bearing captions. It was used for training the model. It differentiates between positive and negative sentiments, so that we have two sections. In the positive section, we find 2873 captions of 998 images for training and 409 captions of 174 images for testing. The negative section includes 2468 of 997 images for training and 1509 captions of 503 images for testing.

B. Evaluation Metrics

Our work was evaluated with common image captioning metrics.

- BLEU [31]: stands for Bilingual Evaluation Understudy Score. It is used to evaluate a generated sentence to a reference sentence. The evaluation is based on the N-gram overlap between machine translation output and reference translation. The N-gram size can be from 1 to 4 and this results in the BLEU-1, BLEU-2, BLEU-3 and BLEU-4 metrics.
- ROUGE-L [32]: ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation and L stands for Longest Common Subsequence based statistics. It is a set of metrics, which compare an automatically produced summary against a human redacted summary taking into account sentence level structure similarity naturally.
- CIDEr [33]: stands for Consensus-based Image Description Evaluation. It is a protocol capturing human consensus. It is meant by consensus how most people tend to describe the input image.

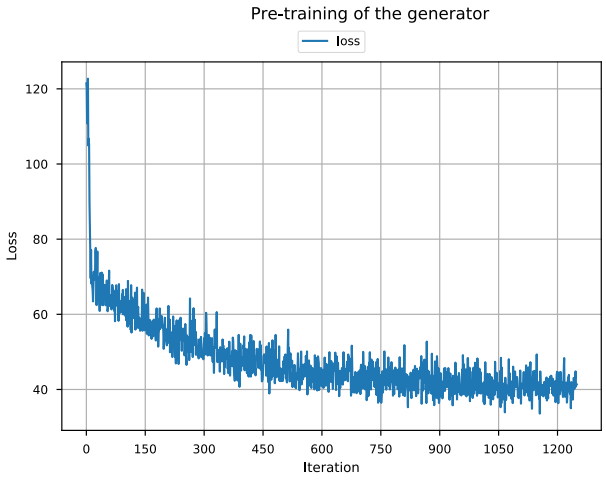

Fig. 8: LSTM-based generator in pre-training mode

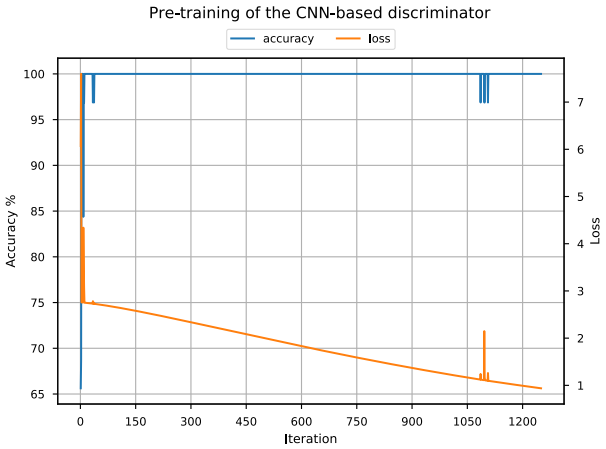

Fig. 9: CNN-based discriminator in pre-training mode

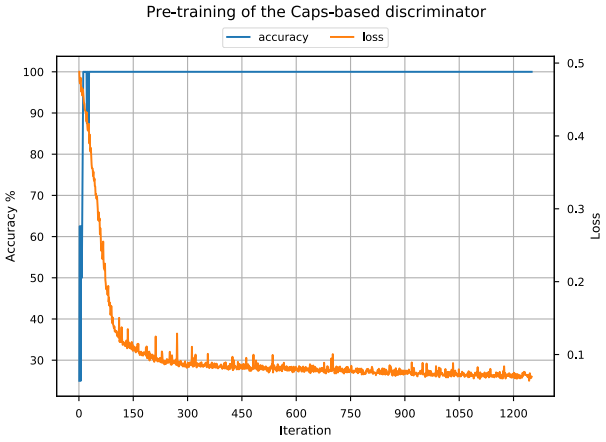

Fig. 10: Caps-based discriminator in pre-training mode

.

[1]https://github.com/ussaema/SeqCapsGAN

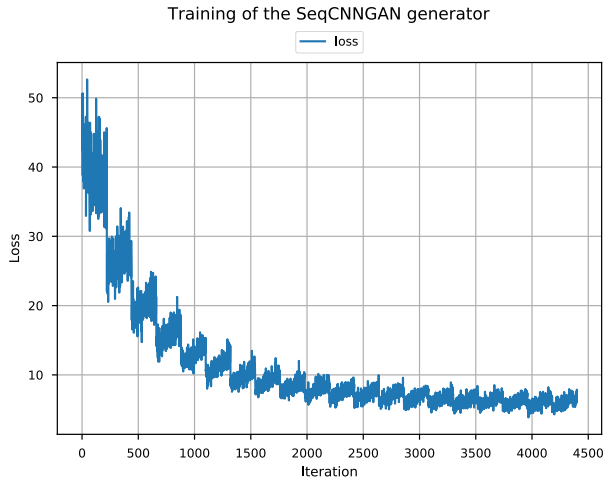

Fig. 11: SeqCNNGAN: LSTM-based generator in adversarial training mode

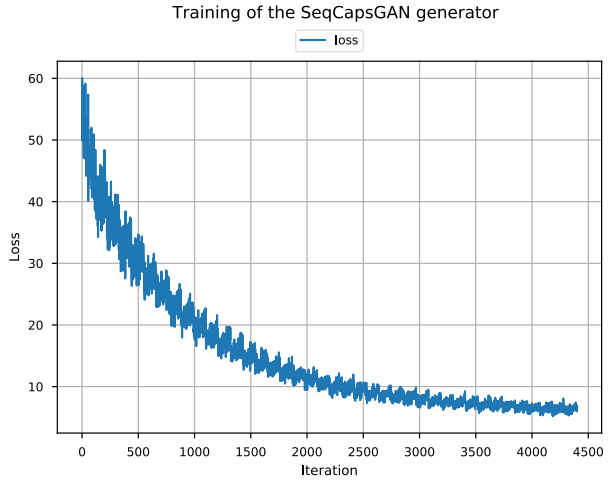

Fig. 12: SeqCapsGAN: LSTM-based generator in adversarial training mode

C. Comparison with baseline models

Pre-training step:

- Behavior of generator: the pre-training on the MS COCO data-set is successful as the loss decreases significantly with small fluctuations (Fig. 8).
- Behavior of discriminator: the loss of the discriminator in the pre-training mode decreases in faster slope using capsules than using CNN. The accuracy is saturated faster in the CNN model but exhibits some fluctuations (Fig. 9 and Fig. 10).

Training step:

- Behavior of generator: the loss function decreases in both models with the same slope. However, the SeqCNNGAN presents more fluctuations and is therefore unstable. (Fig. 11 and Fig. 12)

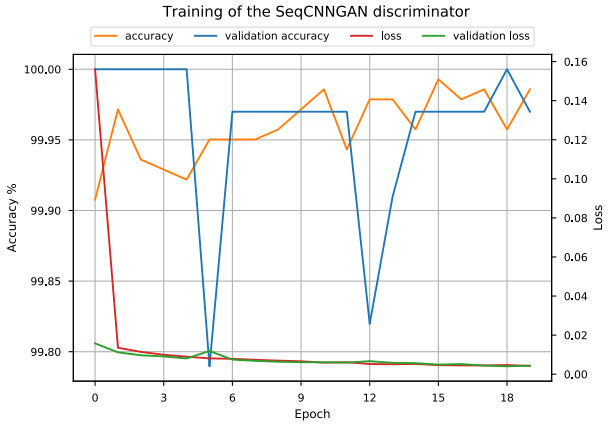

Fig. 13: SeqCNNGAN: Capsule-based discriminator in adversarial training mode

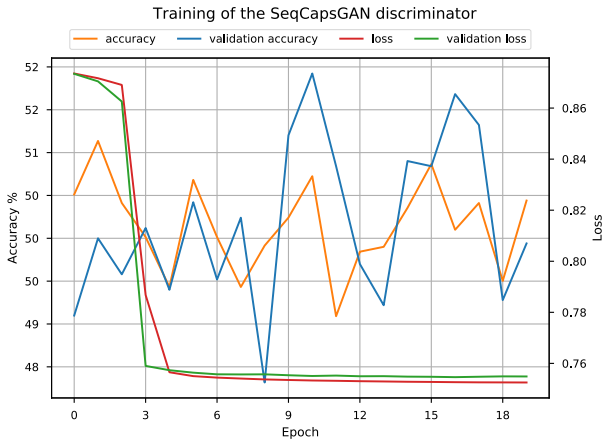

Fig. 14: SeqCapsGAN: Capsule-based discriminator in adversarial training mode

- Behavior of discriminator: Both losses decreases at an early epoch to a minimal value. The accuracy of the CNN-based discriminator is very high and forces the generator to drastically reduce its loss at every epoch beginning. However, the accuracy of the Capsule-based discriminator is varying between the level chance accuracy (50%), which allows the generator to smoothly improve its loss. (Fig. 13 and Fig. 14)

D. Comparison with other works

The Table I presents the performance of previous work based on the BLUE, the ROUGE-L and the CIDEr scores.

The SeqCapsGAN outperforms all other models in terms of BLUE-1 and ROUGE-L scores. In other metrics, the difference to the best model is small.

E. Generated Captions

An observation of sample generated captions by both networks in figure 7 gives us an insight on key characteristics and defects in each model. From the first three examples,

we can infer a few differences between the behavior of the SeqCNNGAN and SeqCapsGAN networks. In figure 7a, the CNN variant still does not meaningfully attribute adjectives to a context for example by the use of the expression "dead grass". In figure 7b, while the Caps-based network was able to learn the right article, the CNN-based one did not learn to adapt it to the context yet. Finally in figure 7c, the SeqCNNGAN model is unable to notice the redundancy between different prepositions in conjunction with lonely road. This is due to the lack of a global observation of the sentence in the CNN variant.

Both models exhibit several failures as in figure 7d, where the network is able to learn the features yet not the right relations. This might be due to the training not including this specific context, especially if the VGGNet did not train for this scenario. In the following example in figure 7e, the description does not match the object in the foreground. In fact, the network eventually learns a wrong feature to word mapping.

V. Conclusion

This work was driven by the lack of an highly end-to-end solution providing accurate sentiment-bearing captions for input images. A performant solution can be based on state-of-the-art techniques in image captioning: GAN networks and text classification CapsNet. This thesis investigates the ability of a GAN network SeqCapsGAN, whose discriminator is composed of CapsNet, to describe visual content in a human-like way.

The implemented model was evaluated with standard image-captioning evaluation metrics against a GAN network with a normal CNN-based discriminator SeqCNNGAN and other baseline models. SeqCapsGAN's performance is comparable with state-of-the-art solutions and it even outperformed them in both BLEU-1 and ROUGE-L metrics.

The main contribution of this work is to provide a new model capable of achieving a considerable performance increase with a faster learning pace and less data.

The network can be further improved especially from the generator side. An interesting work could to try feeding the LSTM network with features extracted from a CapsNet instead of VGGNet features as capsules have shown a stronger ability in encoding visual content.

References

[1] Y. Li, T. Yao, T. Mei, H. Chao, and Y. Rui, "Share-and-chat: Achieving human-level video commenting by search and multi-view embedding," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 928–937.

[2] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "Stylenet: Generating attractive visual captions with styles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3137–3146.

[3] 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.

[4] A. P. Mathews, L. Xie, and X. He, "Senticap: Generating image descriptions with sentiments," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[5] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "Treetalk: Composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351–362, 2014.

[6] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Advances in neural information processing systems*, 2011, pp. 1143–1151.

[7] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, "Collective generation of natural image descriptions," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 359–368.

[8] 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.

[9] 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.

[10] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[11] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille, "Learning like a child: Fast novel visual concept learning from sentence descriptions of images," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2533–2541.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in neural information processing systems*, 2015, pp. 1693–1701.

[14] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom, "Reasoning about entailment with neural attention," *arXiv preprint arXiv:1509.06664*, 2015.

[15] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[16] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.

[17] 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.

[18] Z. Wu and R. Cohen, "Encode, review, and decode: Reviewer module for caption generation," *arXiv preprint arXiv:1605.07912*, 2016.

[19] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[21] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[22] K. Wang and X. Wan, "Sentigan: Generating sentimental texts via mixture adversarial networks." in *IJCAI*, 2018, pp. 4446–4452.

[23] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[25] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *CoRR*, vol. abs/1710.09829, 2017. [Online]. Available: http://arxiv.org/abs/1710.09829

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," vol. abs/1406.2661, 2014. [Online]. Available: http://arxiv.org/abs/1406.2661

[27] Wikipedia, "Zero-sum game," 2016. [Online]. Available: http://en.wikipedia.org/wiki/Zero-sum_game
[28] O. Mohamad Nezami, M. Dras, S. Wan, C. Paris, and L. Hamey, "Towards generating stylized image captions via adversarial training," *arXiv preprint arXiv:1908.02943*, 2019.
[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.
[30] J. HEINONEN, *LECTURES ON LIPSCHITZ ANALYSIS*. [Online]. Available: http://www.math.jyu.fi/research/reports/rep100.pdf
[31] 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.
[32] B. Sharifi, M.-A. Hutton, and J. Kalita, "Summarizing microblogs automatically," in *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 685–688.
[33] 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.