Improving the Naturalness and Diversity of Referring Expression Generation models using Minimum Risk Training

Nikolaos Panagiaris
10 Colinton Road
Edinburgh, EH10 5DT
Edinburgh Napier University
n.panagaris@napier.ac.uk

Emma Hart
10 Colinton Road
Edinburgh, EH10 5DT
Edinburgh Napier University
e.hart@napier.ac.uk

Dimitra Gkatzia
10 Colinton Road
Edinburgh, EH10 5DT
Edinburgh Napier University
d.gkatzia@napier.ac.uk

Abstract

In this paper we consider the problem of optimizing neural Referring Expression Generation (REG) models with sequence level objectives. Recently reinforcement learning (RL) techniques have been adopted to train deep end-to-end systems to directly optimize sequence-level objectives. However, there are two issues associated with RL training: (1) effectively applying RL is challenging, and (2) the generated sentences lack in diversity and naturalness due to deficiencies in the generated word distribution, smaller vocabulary size, and repetitiveness of frequent words or phrases. To alleviate these issues, we propose a novel strategy for training REG models, using minimum risk training (MRT) with maximum likelihood estimation (MLE) and we show that our approach outperforms RL w.r.t naturalness and diversity of the output. Specifically, our approach achieves an increase in CIDEr scores between 23%-57% in two datasets. We further demonstrate the robustness of the proposed method through a detailed comparison with different REG models.

1 Introduction

Referring expression generation (REG) aims at generating utterances that help anchoring an object within an image. Such descriptions are called referring expressions (REs) (Krahmer and van Deemter, 2012). Early work focused on datasets with relatively simple visual stimuli (Viethen et al., 2013; Viethen and Dale, 2010; Mitchell et al., 2013) utilizing synthesized images of objects in artificial scenes. The recently released datasets RefCLEF, RefCOCO, RefCOCO+ and RefCOCOg (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016) which contain natural images of cluttered scenes, led to a surge of interest in using deep neural networks for REG. Such approaches utilize the encoder-decoder paradigm originally proposed for machine translation (Sutskever et al., 2014; Cho et al., 2014) and since have been widely used to various other NLG sub-fields (Tan et al., 2017; Guo et al., 2018; Vinyals and Le, 2015; Li et al., 2016; Vinyals et al., 2015; Xu et al., 2015). The encoder-decoder model consists of a deep convolutional neural network (CNN) (Krizhevsky et al., 2012) to encode the visual features into a fixed-size latent representation, and a variation of recurrent neural network (RNN) (Jain and Medsker, 1999), e.g. a Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) network to generate the output.

The encoder-decoder model is typically trained to maximize the likelihood of a word given the history of generated words so far. This training approach is referred to as “Teacher-Forcing” (Bengio et al., 2015). Although intuitive to train a model on token-level, during generation a model is evaluated based on its ability to optimize towards sequence level metrics resulting in a discrepancy between training and testing objectives. Furthermore, a second problem that stems from “Teacher-Forcing” is that during training, the model uses the ground-truth words to predict the next one, while during testing uses its own predictions. This missmatch, coined as exposure bias (Ranzato et al., 2016), results in error accumulation during generation.

Recently reinforcement learning (RL) (Sutton and Barto, 2018) techniques have been adopted to alleviate the exposure bias problem and directly optimize the non-differentiable task specific metrics. For instance, Ranzato et al. (2016) propose a method that builds upon the REINFORCE algorithm to directly optimize the non-differential test metrics and reports promising results in machine translation, while Bahdanau et al. (2016) utilizes an Actor-critic method, that involves the training of an additional value network to normalize the reward.

However, training with RL is a non-trivial task

Proceedings of The 13th International Conference on Natural Language Generation, pages 41–51,
Dublin, Ireland, 15-18 December, 2020. ©2020 Association for Computational Linguistics

due to a number of limitations: (1) high variance of the gradient (Rennie et al., 2017); (2) lack of per-token advantage, i.e. the REINFORCE algorithm makes the assumption that every token contributes equally to the whole sequence (Wu et al., 2016); and (3) reward configuration (Bahdanau et al., 2016; Ranzato et al., 2016). Furthermore, effectively applying RL to REG has not been explored, with the exception of (Yu et al., 2017) who incorporate an additional module to reward discriminative REs by updating the speaker with a policy gradient algorithm. However, little is reported of how the RL was configured. To the best of our knowledge, this is the first work to thoroughly propose how to effectively train REG models with RL.

Furthermore, beside the aforementioned limitations of RL methods there is another problem that is often overlooked. While directly optimizing the evaluation metrics one can achieve higher scores, the generated text lacks diversity due to repeated n-grams (Wang and Chan, 2019). Our analysis shows that RL trained models are strongly biased towards frequent REs leading to smaller vocabulary and deficiencies in the generated word distribution.

To address these issues we propose the use of minimum risk training (MRT) (Och, 2003) as an alternative way of optimizing REG systems on sequence level. Minimum risk training aims at minimizing the expected loss over training data by taking automatic evaluation metrics into consideration. The MRT objective has the following advantages over MLE. First, it can directly optimize sequence level objectives that are not necessarily differentiable. Second, while MLE maximizes the likelihood of the training data, MRT introduces a notion of ranking amongst candidate sequences by discriminating between sequences. Thus, by minimizing the risk, we expect to find a distribution that approximates well the ground-truth distribution. Furthermore, the MRT objective is similar to the REINFORCE algorithm in a sense that both maximize an expected reward or cost. However, there are two fundamental advantages of the MRT over RL: (1) the REINFORCE algorithm typically utilizes one sample in order to approximate the expectation, whereas the MRT objective considers multiple sequences making it sample and data sufficient; and (2) the MRT objective intuitively estimates the expected risk over a set of candidate sequences, whereas the REINFORCE algorithm typically relies on the baseline reward to determine effectively the sign of the gradient.

Therefore, our main contributions are as follows: Firstly, we conduct an extensive analysis and benchmarking of RL training strategies for REG, by exploring how different aspects such as the reward and the baseline reward configuration affect REG models (Section 8.1). Our experiments reveal how to best train REG models using reinforcement learning. Secondly, we show that models optimised for CIDEr also achieve higher scores in all other metrics (BLEU etc.) even when compared to models directly optimised on them. Although our RL approach outperforms the state-of-art, RL still suffers from the limitations discussed earlier. Therefore, we propose a novel training strategy for REG which combines MRT with MLE and we show its effectiveness in comparison to a number of RL training strategies w.r.t naturalness, diversity and informativeness (Section 8.2). Our approach achieves improvements between 33.5%-38.7% and 23.4%-57.8% in terms of CIDEr on RefCOCO and RefCOCO+ respectively compared to previously proposed approaches. Finally, a detailed analysis shows that when a REG model is trained with the proposed approach, uses a larger vocabulary, produces longer referring expressions and generates more uni-grams and bi-grams.

2 Related Work

Early work in referring expression generation can be dated back to the early 1970s (Winograd, 1972). The traditional view of REG is a two step procedure where the REG model accounts for the content selection and determination of the referential form (Krahmer and van Deemter, 2012). However, the large body of work in REG focuses on the determination of content for definite descriptions (Krahmer and van Deemter, 2012). Algorithms such as the *full brevity* and the *incremental algorithm* (Dale and Reiter, 1995) have as foundation the Gricean maxims (Grice, 1975), that provide insights of how people behave in different communication scenarios (Krahmer and van Deemter, 2012).

Recently due to the availability of larger and more complex natural image datasets, such as RefCOCO (Yu et al., 2016; Mao et al., 2016) there is a surge of interest in applying deep learning methods. Neural REG approaches rely on incorporating contextual information by using visual features, appearance attributes (Liu et al., 2017), location features (Yu et al., 2016) and global image features as target object representation. In their seminal work, Mao et al. (2016) use a convolutional neural network to extract visual features and an LSTM to generate the expression trained on Maximum Mutual Information objective. Yu et al. (2016) propose a unified framework where a speaker module generates REs, a listener module comprehends REs, and a reinforcer module provides guidance towards informative REs. Lastly, Zarrieß and Schlangen (2018) examine the impact that variations of beam search have in the length of REs.

Although there are not published attempts on optimizing neural REG systems on a sequence level, we will review a number of works from the wider field of natural language generation. Ranzato et al. (2016) were the first to adopt the REINFORCE algorithm in order to optimize the encoder-decoder model. The discovery that baselines can effectively reduce the variance of the gradient estimation led to a significant body of work in NLG. Murphy et al. (2017) used fully connected layers to predict the baseline and used Monte Carlo rollouts to approximate the state-action value. Bahdanau et al. (2016) utilize an actor-critic framework and combine it with temporal difference learning. The state-action value was modelled by a separate RNN. Rennie et al. (2017) propose the utilization of the output of the model at the test time to normalize the reward. Although MRT has a long history in training linear model for structured predictions, it has only be used in neural machine translation (Shen et al., 2016; Edunov et al., 2018) as an alternative to MLE training. In this work, however, we apply MRT to REG as an alternative to RL and we compare the output of those two training strategies in terms of naturalness and diversity.

3 REG model

As this work focuses primarily on the training objectives for neural REG models, we adopt a standard encoder-decoder architecture language model similar to (Rennie et al., 2017; Vinyals et al., 2015). The encoder is a CNN network that extracts the representation of the target object. Then this representation is embedded through a linear projection layer W_I. The words are represented as one-hot vector, projected to the same space as the visual representation through a linear embedding layer. The start of each sequence is denoted by a special **BOS** token, while the special stop token **EOS** denotes the end of the sequence. The decoder, which is responsible for the generation of REs is modeled as an LSTM network. The image features are used only as an input to $t = 0$ in order to initialize the LSTM based on the visual contents. Then, at each time step t, its output depends on the previously generated words and on the hidden state, which encodes the knowledge of the observed input up to this time step.

The parameters θ of the model are learned by maximizing the likelihood of the observed sequence. Specifically, given N training pairs the training objective is defined as:

$$\begin{aligned}\mathcal{L}_\theta &= \frac{1}{N}\sum_{n=1}^{N}\log p(y^n|o^n, I^n\theta)\\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\log p(y_t^n|y_{1:t-1}^n, o^n, I^n\theta)\end{aligned}\tag{1}$$

where o^n is the n'th object in the I^n image, $y^n = (y_1^n, \ldots, y_{T_n}^n)$ is the ground truth referring expression of the n'th object and N is the total number of training examples.

4 Training REG with Reinforcement Learning

The generation process can be cast into a reinforcement learning process as first described in (Ranzato et al., 2016). Within the classic reinforcement learning paradigm, an agent performs an action under a specific policy π. The nature of the policy is application dependent. Within REG the language model can be seen as an agent that interacts with its environment (i.e. the previously generated words and the visual features at each time step t). The parameters θ of the agent define a policy π_θ. The agent selects an action, which is a candidate token from the vocabulary under the policy, until it generates the EOS token. Once the agent reaches the end of the sequence it observes a terminal reward r, which is the score for generating a RE $\hat{y}_n$ given an object o and a ground truth referring expression (or a set of referring expressions) y. The reward is a scalar produced by any evaluation metric such as CIDEr. Therefore, the training aims at parameterizing the agent in order to maximize the reward as follows:

$$\begin{aligned}\mathcal{L}_\theta &= \sum_{n=1}^{N} E_{\hat{y}\sim\pi_\theta(\hat{y}_n|o^n,y^n)} r(\hat{y}_n|o^n, y^n)\\ &= \sum_{n=1}^{N}\sum_{\hat{y}\in Y}\pi_\theta(\hat{y}|o^n, y^n) r(\hat{y}_n|o^n, y^n),\end{aligned}\tag{2}$$

where N is the number of examples in the training set and Y denotes the entire space of all possible output referring expressions, which is intractable to enumerate or score with a model. Instead, REINFORCE allows to optimize the gradient of the expected reward by sampling $\hat{y}$ from the policy $p(y|o,)$ during training. Thus, it aims to maximize the following objective:

$$\mathcal{L}_\theta = \sum_{n=1}^{N} r(\hat{y}^n|o^n, y^n), \hat{y}^n \sim \pi_\theta(\hat{y}|o^n, y^n) \quad (3)$$

An inherent challenge of the REINFORCE algorithm is that typically leads to highly unstable training due to the noise in gradient estimation and reward computation (Rennie et al., 2017; Ranzato et al., 2016). Thus, in the next sections we explore a number of methods that have been proposed in literature that stabilize training. Specifically, for reward computation we study: (1) how to sample the candidate samples; (2) which reward function to use; and (3) whether reward sampling is beneficial. Furthermore, as a variance reduction technique we explore the applicability of self-critical training proposed for image captioning (Rennie et al., 2017). Lastly, we explore whether the combination of MLE training with RL improves the diversity and naturalness of the output.

4.1 Reward Configuration

The standard training for REG poses two uncommon challenges for RL. First, the action space in REG problems is a high-dimensional discrete space that it is intractable, while in the classic RL paradigm the common scenario is a smaller discrete action space (e.g. games (Mnih et al., 2015)), or a relatively low dimension continuous space of actions (e.g. robotics (Lillicrap et al., 2016)). Hence, the first important factor is the *search strategy* for generating the sequence of actions. Secondly, the reward for REG is naturally sparse since each token of the training sequence is assigned the same reward value. Note that, the reward is observed when the full sequence is produced. Thus, we explore whether a cumulative reward is better than the terminal reward. This process is known as reward shaping (Ng et al., 1999).

We consider two search strategies for generating sequences. The first is *beam search*, that finds the most likely sequence by performing a greedy breadth-first search over a limited search space. Specifically, each candidate sequence is expanded from left to right selecting all possible tokens from the vocabulary at a time. From this set, the $top - k$ candidate sequences with the highest probabilities are selected, and the beam search process continues until the $top - k$ candidates with the highest probability are returned. The second strategy is random sampling, which randomly samples from the model's distribution at every time-step until the end of the sequence token is produced.

Balancing between *exploration* and *exploitation* is a major challenge in RL. For instance, it may be required for an agent to pick an action associated with the highest expected reward (i.e. exploitation). However, in this scenario it may fail to learn more rewarding actions. Therefore exploration, that is the choice of new actions and the visit of new states, may also be beneficial. Beam search focuses on producing high probability sequences and therefore is considered as an exploitation strategy, while random sampling introduces more diverse sequences and thus contributes towards the exploration of the action states. However, due to the fact that the actions are being sampled from the model being optimized the exploration is de facto limited.

Although we aim to optimize a REG system to produce sequences that maximize a sequence level metric, simply awarding this score at the last step of a complete episode (sequence generation) provides naturally a sparse training signal. An agent however picks a number of actions in order to produce a sequence (dependent on the length of the sequence). In other words, assigning a terminal reward to the entire sequence is equivalent to a uniform token-level reward. Dense rewards can be easier to learn from, thus we explore the use of reward shaping (Ng et al., 1999) as proposed in (Bahdanau et al., 2016). Specifically, given the sequence of actions (i.e. words) $y_1...y_{t-1}$ executed by the agent until time step t, the intermediate reward is calculated as: $r_t(\hat{y}_t, y) = r(\hat{y}_{1...t}, y) - r(\hat{y}_{1...t-1}, y)$ by comparing the incomplete sequence with the ground truth. Thus, at time step t the model's parameters are updated based on the cumulative reward.

4.2 Variance Reduction with Self-Critical Training

Another important weakness of the REINFORCE algorithm is that it exhibits high variance that leads to unstable training without proper context-dependent normalization. An intuitive way to reduce the variance is to reduce the magnitude of the learning signal by subtracting a quantity, called a *baseline*. It can be any value as long as it is independent of the parameters of the agent. For instance, one can sample N sequences of actions and update the gradient by averaging over the N sequences. In this case, the baseline could be the mean of the rewards of the N sequences.

As another solution to reduce the variance of the gradient estimator, Rennie et al. (2017) proposed a self-critical training scheme. In order to calculate the baseline reward under this training strategy, two independent sequences are produced: $\hat{y}$, which is obtained by sampling from the policy, and $\widehat{y^g}$, the baseline output, obtained by performing greedy search. Thus, the training aims to minimize the following objective:

$$\mathcal{L}_\theta = \sum_{n=1}^{N}(r(\hat{y}|o^n, y^n) - r(\hat{y}^g|o^n, y^n)) \tag{4}$$

The minimization of L_θ is analogous of maximizing the conditional likelihood of the sampled sequence $\hat{y}$ if it obtains a higher reward than the baseline $\hat{y}^g$, thus increasing the reward expectation.

5 Minimum Risk Training for Referring Expression Generation

Beside the aforementioned problems, there are two other limitations that are often overlooked. First, while these methods can directly optimize the non-differentiable rewards and improve the performance of evaluation metrics, the generated text suffers from lack of diversity due to repetition of common n-grams. The second limitation is that the approximation of the reward is based on one sample which is data and sample inefficient. To address these limitations we explore a principled alternative to the REINFORCE algorithm, the minimum risk training (Och, 2003).

Minimum risk training (MRT) minimizes the value of a given task-specific cost function, i.e. risk, over the training data at sequence level. Specifically, let x denote a fixed-size representation of the input, then the set $\mathcal{Y}(\mathbf{x}^{(\mathbf{s})})$ denotes the set of all possible referring expressions generated by the model with parameters θ. For a given candidate sequence $\mathbf{y}'$ and ground truth referring expression $\mathbf{y}$, MRT defines a cost function $\Delta(\mathbf{y}', \mathbf{y})$ which is the semantic distance between $\mathbf{y}'$ and the standard $\mathbf{y}$. The cost function can be any function that captures the discrepancy between the model's prediction and the ground truth. Formally, the objective function of MRT is the following:

$$\mathcal{L}_{\text{MRT}} = \sum_{n=1}^{N} \mathbb{E}_{\mathcal{Y}(\mathbf{x})}\Delta(\mathbf{y}', \mathbf{y}^{(\mathbf{n})}). \tag{5}$$

where $\mathbb{E}_{\mathcal{Y}(\mathbf{x})}$ denotes the expectation over the set of all possible candidate sequences $\mathcal{Y}(\mathbf{x}^{(\mathbf{n})})$. However, as previously mentioned enumerating and scoring candidate sequences over the entire space is intractable. Instead, we sample a subset $\mathcal{S}(\mathbf{x}) \subset \mathcal{Y}(\mathbf{x})$ to approximate the probability distribution, and formalize the objective function as:

$$\mathcal{L}_{\text{MRT}} = \sum_{s=1}^{S}\sum_{\mathbf{y}'\in\mathcal{S}(\mathbf{x}^{(\mathbf{s})})} \frac{p(\mathbf{y}'|\mathbf{x}^{(\mathbf{s})})}{\sum_{\mathbf{y}^*\in\mathcal{S}(\mathbf{x}^{(\mathbf{s})})} p(\mathbf{y}^*|\mathbf{x}^{(\mathbf{s})})}\Delta(\mathbf{y}', \mathbf{y}^{(\mathbf{s})}) \tag{6}$$

The MRT objective minimizes the expected value of a cost function which enables us to optimize REG models with respect to specific evaluation metrics of the task. In this work we explore the use of various REG evaluation metrics such as CIDEr and BLEU and combination of those. Furthermore, for the construction of the subset of the candidate sequences we consider online setting, specifically we regenerate the candidate set for each training sample. Again we consider random sampling and beam search as search strategies (see Section 4.1). Moreover, we also considered offline generation, that is the candidate sequences are generated before training and never refreshed. However, we found that it leads to inferior performance and thus was not included.

6 Combined objectives

We also experiment with combining the MLE training objective either RL or MRT. The motivation of the loss combination is to maintain good token-level accuracy while optimizing on the sequence-level. In other words, using an evaluation metric as a reward can suppress the probability of the words that do not increase the metric score, and thus concentrate the distribution to a single point. Thus, we explore a combined objective in order to scale the peakiness of the output distribution. Specifically, the weighted combination of MLE (Equation 1) with RL objective (Equation 4) is defined as follows:

$$L_{weighed_{RL}} = (1 - \alpha) * L_{mle} + \alpha * \hat{L}_{rl}, \tag{7}$$

Equivalently, combing the MRT objective (Equation 6) with MLE we have:

$$L_{weighed_{MRT}} = (1 - \alpha) * L_{mle} + \alpha * \hat{L}_{MRT}, \tag{8}$$

where α is a scaling factor controlling the difference in magnitude between the combined objectives.

7 Experimental Setup

7.1 Datasets

We trained our models on RefCOCO and RefCOCO+ (Yu et al., 2016). Although both datasets contain similar images since they are built upon the MSCOCO dataset (Lin et al., 2014), the textual properties of their expressions are different due to different data collection objectives. In particular, for ReFCOCO+, the use of absolute location words (e.g. top right, bottom left, etc.) was not allowed and thus the RE are *appearance* focused, while for the RefCOCO the use of *location* is essential in order for the target object to be successfully individualized. Furthermore, for each dataset different test splits are provided. The predefined test splits for both datasets are divided between person vs object splits. In particular, images containing people are in "TestA" and images that contain all other object categories are in "TestB".

7.2 Implementation Details

Visual Features The visual representation used is a 4101-dimensional vector that is a concatenation of: (1) a 2048-dimensional vector of the target object region; (2) a 2048-dimensional vector representation of the whole image that serves as context features and (3) object location features as presented in (Yu et al., 2016). As main feature extractor we used ResNet-152 (He et al., 2016). In more detail, for the object region features, the aspect ratio of the region was kept constant and was scaled to 224 × 224 resolution. The margins were padded with the mean pixel value, following (Mao et al., 2016).

Training For our language model, we set the dimension of LSTM hidden state, image feature embeddings, and word embeddings to 512. The batch size is set to 128 images. The learning rate is initialized to be 5×10^{-4}, and then annealed by shrinking it by a factor of 0.8 for every three epochs. Both the RL and MRT models are trained according to the following scheme: We first pretrain the language model using MLE, optimized with Adam (Kingma and Ba, 2014). At each epoch, we evaluate the model on the validation set and select the model with the best CIDEr score as an initialization for RL and MRT training. We then run RL or MRT training initialized with the MLE model to optimize the CIDEr metric using ADAM with a fixed learning rate 5×10^{-5}.

7.3 Evaluation

For evaluation we opt for automatic metrics. Specifically, in order to measure the naturalness of referring expressions we use the standard automatic metrics that have been used in REG (Mao et al., 2016; Zarrieß and Schlangen, 2018; Yu et al., 2016) that compare the generated referring expression with the human ones: $BLEU_1$ for unigrams, CIDEr and METEOR. In order to evaluate the diversity, we report: (1) the average length of referring expressions (ASL) (2) the number of unique words of the generated corpus; (Voc) and (3) the average number of unique bigrams per 1000 bigrams (TTR). (van Miltenburg et al., 2018).

8 Results & Discussion

8.1 Evaluating different RL training strategies

We first explore a number context-dependent normalization factors that affect the RL training described in Section 4. Regarding the reward configuration (see Section 4.1) we explore: (1) which reward function to use to evaluate the sequences; (2) which search strategy will be used to sample the actions from the policy; and (3) whether reward normalization further stabilizes the training.

Reward Function: First we compare various evaluation measures as reward functions, namely CIDEr, BLEU and METEOR as well as metrics combinations. A summary of the results is given in Table 1, where RL stands for the REINFORCE algorithm. We present the performance of the MLE model we used for the initialization of the RL training. As expected, optimizing towards a particular evaluation metric during training leads to an increase on that particular metric during testing. However, the benefits are not comparable with those gained when optimizing CIDEr. Specifically, CIDEr optimization leads to improvements in scores for all other metrics as opposed to directly optimize them. A notable exception is the combination of CIDER+BLEU where BLEU score is higher compared to optimizing only for CIDEr. Therefore, for the rest of the paper, all RL models are based on CIDEr optimization.

	testA			testB		
Method	BLEU	METEOR	CIDEr	BLEU	METEOR	CIDEr
MLE	0.542	0.200	0.841	0.614	0.258	1.507
RL + CIDER	0.569	**0.222**	**0.954**	**0.625**	**0.277**	**1.564**
RL + BLEU	0.557	0.210	0.860	0.616	0.261	1.510
RL + METEOR	0.533	0.205	0.834	0.586	0.260	1.508
RL + CIDER + BLEU	**0.579**	0.221	0.945	**0.625**	0.271	1.532
RL + CIDER + METEOR	0.563	0.219	0.947	0.607	0.266	1.523

Table 1: Performance of different reward functions on RefCOCO dataset (the same trend applies to RefCOCO+ and thus omitted). RL stands for the REINFORCE algorithm. Optimizing the training for the CIDEr metric increases all evaluation metrics significantly. All models were decoded using greedy decoding. The performance of the seed model is also reported. The best overall values for each metric are emphasized with bold.

	testA			testB			testA+			testB+		
Method	BLEU	METEOR	CIDEr	BLEU	METEOR	CIDEr	BLEU	METEOR	CIDEr	BLEU	METEOR	CIDEr
MLE	0.542	0.200	0.841	0.614	0.258	1.507	**0.481**	0.179	0.715	**0.409**	**0.173**	0.829
RL+ RS	0.569	0.222	0.954	0.625	0.277	1.564	0.469	0.185	0.745	0.286	0.163	0.913
RL + BS	0.561	0.217	0.946	0.617	0.270	1.549	0.465	0.184	0.743	0.277	0.160	0.901
RL + RS+ Shaping	0.574	0.223	0.957	0.628	0.278	**1.567**	0.473	0.181	0.752	0.279	0.162	0.915
RL + BS+ Shaping	0.565	0.219	0.948	0.618	0.272	1.552	0.468	0.155	0.749	0.275	0.161	0.904
SCTS+RS	**0.593**	**0.231**	**1.012**	**0.638**	**0.290**	**1.607**	**0.481**	**0.194**	**0.809**	0.282	0.165	**0.942**
SCTS+GD	0.583	0.227	0.995	0.635	0.279	1.585	0.461	0.185	0.761	0.276	0.163	0.934

Table 2: Results of different search strategies for reward computation and variance reduction. “RS” stands for random sampling, while “BS” refers to beam search and “GD” for greedy decoding. “SCTS” refers to self-critical training. Shaping denotes that we used reward shaping.

Action sampling strategy: So far we sampled the words using random sampling. Next, we compare beam search and random sampling as search strategies to sample the words. The results are shown in Table 2. Although beam search (with width of 2) has been the de facto decoding strategy for neural REG systems, it produces inferior results when compared to random sampling. We hypothesize due to the deterministic nature of beam search, the sampled sequences are often duplicates and thus uninformative for the gradient estimation, while the stochasticity of sampling generates sequences with exploratory usefulness for the gradient estimation and it results in more natural-sounding expressions.

Self-critical training for REG: We next investigate, whether the inclusion of a baseline is an effective way of stabilizing the training by reducing the variance of the gradient. We follow the self-critical training strategy that utilizes the output of the greedy decoding to normalize the rewards. We further investigate random sampling and greedy decoding as search strategies. Table 3 depicts the results. Self-critical training improves over the REINFORCE algorithm, which indicates that the variance of the gradient is significant in neural REG. However, we notice that instead of using the greedy decoding that is originally proposed in (Rennie et al., 2017) random sampling is a better choice.

Combining MLE with RL: Next we evaluate the combination of self-critical objective with MLE. Figure 1 shows the results on the validation set. The best trade-off between MLE and RL objectives in our experiment is when $\alpha = 0.9$. Table 3 depicts the results on the test set where we observe that the weighed combination of MLE and SCTS objective further improves the quality of the generated expressions.

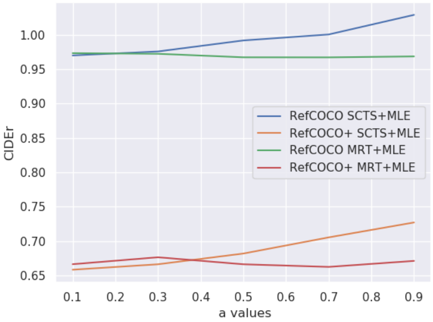

Figure 1: Validation set CIDEr scores for different values of α for combining MLE with either RL objective (see Equation 7) or MRT objective (see Equation 8). Best viewed in color.

8.2 Evaluating Minimum Risk Training for REG

In this subsection, we report the results for training a REG model with minimum risk training and we compare it with MLE. Training with MRT requires generating and scoring multiple candidate referring expressions for each input. Thus, we explore two factors: (1) which search strategy should be used

	testA					testB					testA+					testB+				
Method	CIDEr	BLEU	ASL	Voc	TTR	CIDEr	BLEU	ASL	Voc	TTR	CIDEr	BLEU	ASL	Voc	TTR	CIDEr	BLEU	ASL	Voc	TTR
MLE	0.841	0.542	2.685	121	0.28	1.507	0.614	2.65	158	0.443	0.715	0.481	2.50	148	0.289	0.829	0.409	3.20	229	0.458
SCTS	1.012	0.593	2.610	76	0.20	1.607	0.638	2.37	104	0.388	0.809	0.481	2.47	73	0.196	0.942	0.282	1.44	99	0.520
MRT	0.952	0.569	2.476	124	0.32	1.614	0.632	2.35	161	0.473	0.773	0.483	1.92	127	0.296	0.935	0.413	2.21	217	0.548
MLE+ SCTS	1.032	0.593	2.693	100	0.24	1.700	0.658	2.43	128	0.431	0.816	0.483	2.36	91	0.224	**0.975**	0.302	1.49	132	0.570
MLE+MRT	**1.075**	**0.603**	**2.640**	**141**	**0.33**	**1.763**	**0.692**	**2.73**	**198**	**0.494**	**0.821**	**0.513**	**2.77**	**176**	**0.337**	0.907	**0.430**	**3.37**	**288**	**0.708**
(Yu et al., 2017)	0.775	-	-	-	-	1.320	-	-	-	-	0.520	-	-	-	-	0.735	-	-	-	

Table 3: System results: CIDEr and BLEU scores; average sentence length (ASL); vocabulary size (Voc); mean-segmented bigram ratio (TTR); SCTS denotes self-critical training with random sampling as baseline; MRT denotes minimum risk training with candidate size of 5 for RefCOCO and size of 8 for ReFCOCO+.

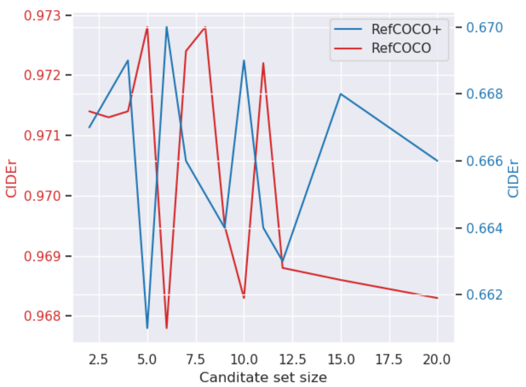

Figure 2: Validation set CIDEr scores for different candidate set sizes for the MRT model. Best viewed in color.

to generate the candidate sequences; (2) and how many sequences we should generate for one input. We found that random sampling performs better than beam search both in terms of CIDEr score and is considerably faster. Thus, Figure 2 compares different set sizes on the validation set when random sampling is used. For RefCOCO we choose candidate set size of 5, while for RefCOCO+ 8. Table 3 presents the results on the test set. Optimizing the REG model with MRT improves both CIDEr and BLEU by several figures over the MLE.

8.3 Comparison of MRT to RL training

Our final experiment compares MRT to RL training w.r.t naturalness and diversity. Table 3 shows all sequence level optimization methods used. When analyzing the effect that different training methods have on naturalness and diversity of the referring expressions a few clear patterns can be observed: (1) SCTS has the lowest diversity and naturalness (i.e. BLEU score) and highest repetition among all models; (2) Out of the 4 different test sets, SCTS has the highest accuracy CIDEr scores when compared to MLE and MRT training; (3) combining the SCTS loss with MLE improves sightly the accuracy, naturalness and diversity of the produced referring expressions. Still, however, the diversity is considerably lower than MLE and MRT. (4) Minimum risk training improves over MLE in all tests sets. However, when compared to SCTS it only produces higher CIDEr in only one case (i.e. RefCOCO testB); (5) MRT has the highest diversity and naturalness compared to the other two training strategies; (6) combing the MRT loss with MLE further improves the diversity and naturalness of the generated referring expressions. In particular, as can be seen in Table 3, the MLE + MRT loss achieves the highest scores in all categories, except in testB+ where the combination of two losses produces inferior results in terms of CIDEr.

Examples of generated REs are illustrated in Figure 3. In all images presented in Figure 3, we observe that the proposed MLE + MRT model improves over all compared training objectives in inferring more pragmatically adequate referring expressions by using, for example, precise appearance and location attributes (e.g. "man with hand on chin" and "left side of pic brown thing in front") or negations (e.g "cat no reflection")

9 Conclusion

In this work we considered the problem of optimizing referring expression generation models with sequence level objectives. Specifically, we firstly provide a comprehensive comparison of different aspects of configuring REG models with RL training. We found that (1) random sampling is a better search strategy than beam search; (2) we showed that using random sampling with self-critical training improves CIDEr scores; (3) incorporating reward shaping improves the performance; (4) we showed that combining RL objectives with MLE is beneficial to the training, resulting in higher CIDEr scores and diversity. However, there is a considerable gap between MLE and RL methods w.r.t. to diversity. Thus, as an alternative to RL we proposed the use of minimum risk training. We showed that MRT combined with MLE produces superior re-

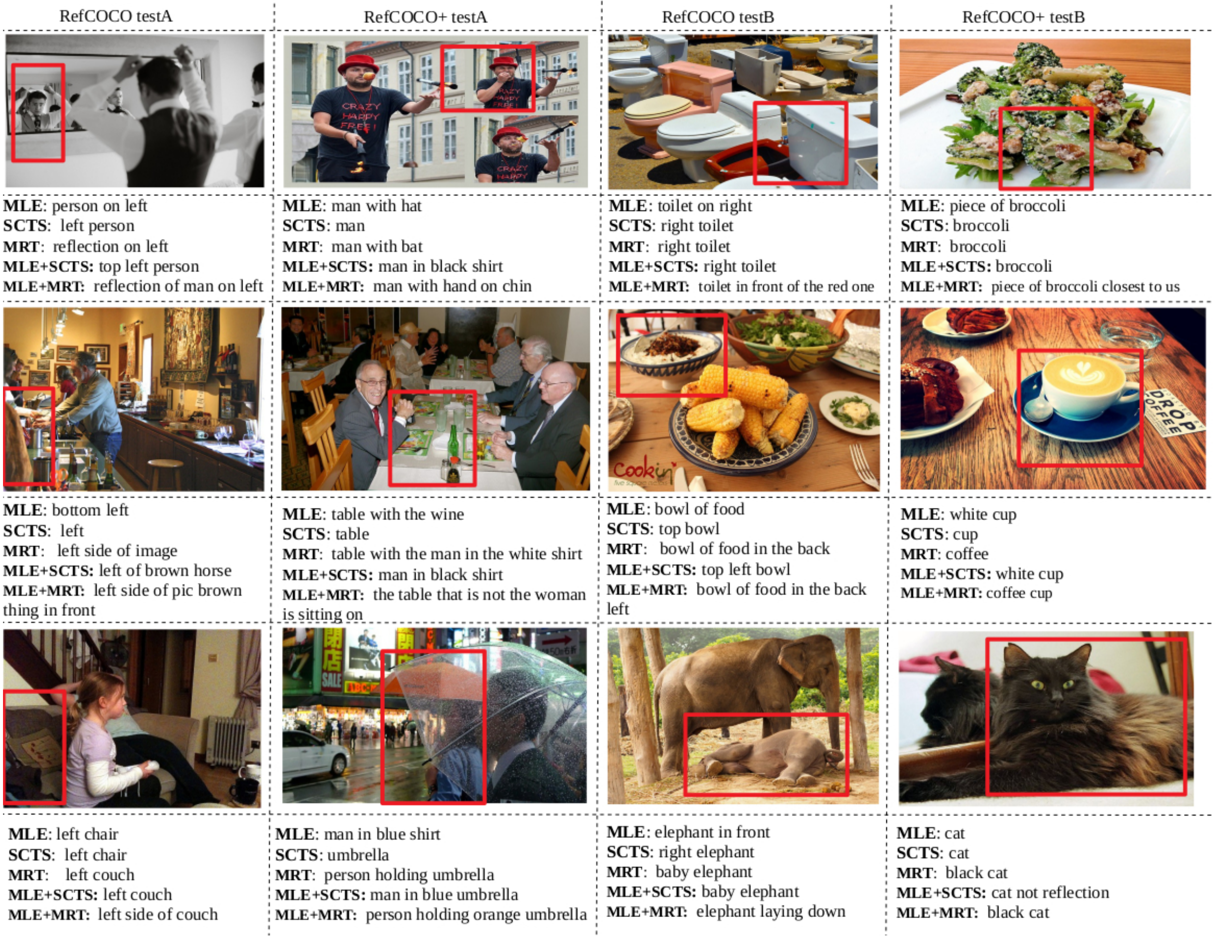

Figure 3: Examples of objects and expressions drawn from both RefCOCO and RefCOCO+ datasets. The target object is highlighted with a red box.

sults in terms of naturalness and diversity of the referring expressions compared to both MLE and RL training. While we have focused on analyzing the performance of the presented models with automated evaluation metrics, we intend to further verify these results in a human evaluation.

10 Acknowledgements

We wish to thank NVIDIA for its kind donation of the GPU used in the presented experiments. DG is supported under the EPSRC project CiViL (EP/T014598/1).

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv e-prints*, abs/1607.07086.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Herbert P. Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and*

Semantics: Vol. 3: Speech Acts, pages 41–58. Academic Press, New York.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Lakhmi C. Jain and Larry R. Medsker. 1999. *Recurrent Neural Networks: Design and Applications*, 1st edition. CRC Press, Inc., Boca Raton, FL, USA.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Comput. Linguist.*, 38(1):173–218.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th Conference on Advances in Neural Information Processing Systems*.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *ICLR*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV*.

Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. 2017. Referring expression generation and comprehension via attributes. In *The IEEE International Conference on Computer Vision (ICCV)*.

J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2018. Measuring the diversity of automatic image descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1730–1741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Margaret Mitchell, Kees van Deemter, and Ehud Baruch Reiter. 2013. Generating expressions that refer to visual objects. In *Proc of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Kevin Murphy, Ning Ye, Sergio Guadarrama, Siqi Liu, and Zhenhai Zhu. 2017. Improved image captioning via policy gradient optimization of spider.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations ICLR*.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jette Viethen and Robert Dale. 2010. Speaker-dependent variation in content selection for referring expression generation. In *Proceedings of the Australasian Language Technology Association Workshop 2010*, pages 81–89, Melbourne, Australia.

Jette Viethen, Margaret Mitchell, and Emiel Krahmer. 2013. Graphs and spatial relations in the generation of referring expressions. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 72–81, Sofia, Bulgaria. Association for Computational Linguistics.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qingzhong Wang and Antoni B. Chan. 2019. Describing like humans: On diversity in image captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Terry Winograd. 1972. *Understanding Natural Language*. Academic Press, Inc., USA.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling Context in Referring Expressions. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*.

Sina Zarrieß and David Schlangen. 2018. Decoding strategies for neural referring expression generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 503–512, Tilburg University, The Netherlands. Association for Computational Linguistics.